Space-efficient population protocols for exact majority on general graphs

Joel Rybicki, Jakob Solnerzik, Olivier Stietel, Robin Vacus

Humboldt University of Berlin

January 11, 2026

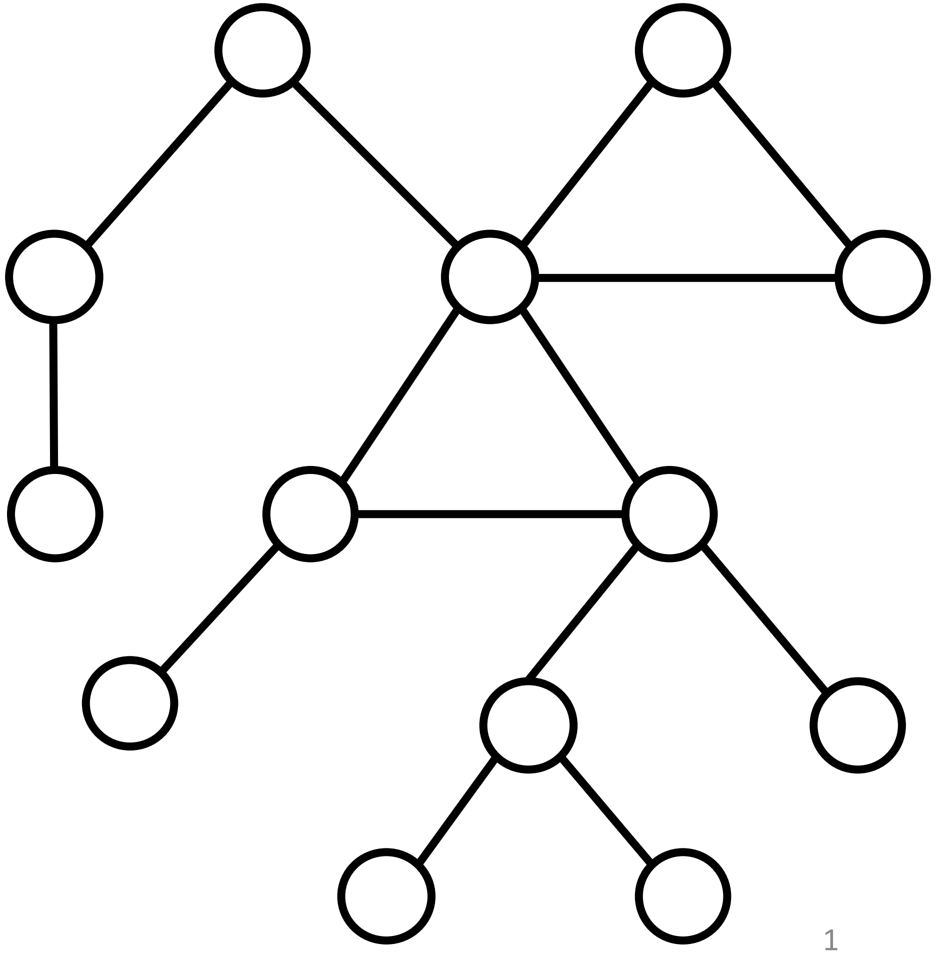
Population protocols & exact majority

Population protocols & exact majority

- Distributed setting
  - Multiple agents (finite state automatons)
  - Each agent runs the same algorithm (protocol)

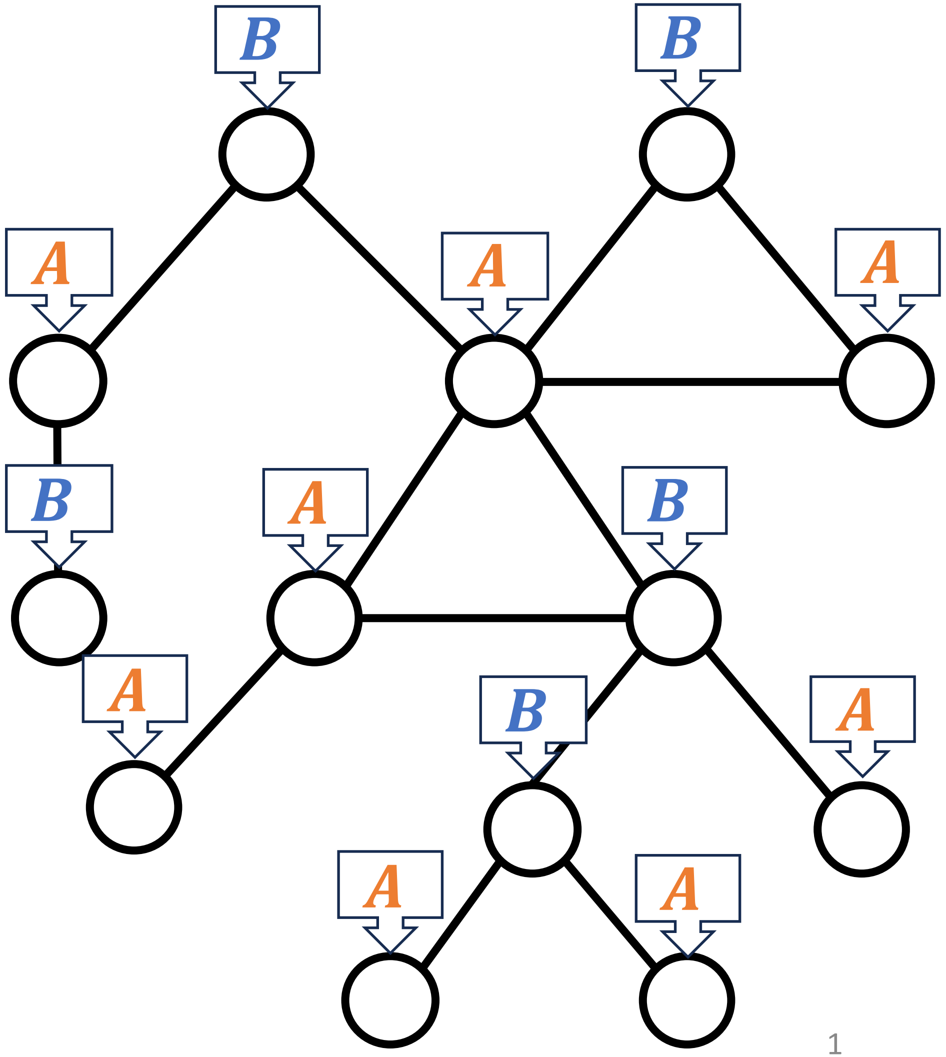
Population protocols & exact majority

- Distributed setting
  - Multiple agents (finite state automatons)
  - Each agent runs the same algorithm (protocol)
- Given:
  - Undirected graph G



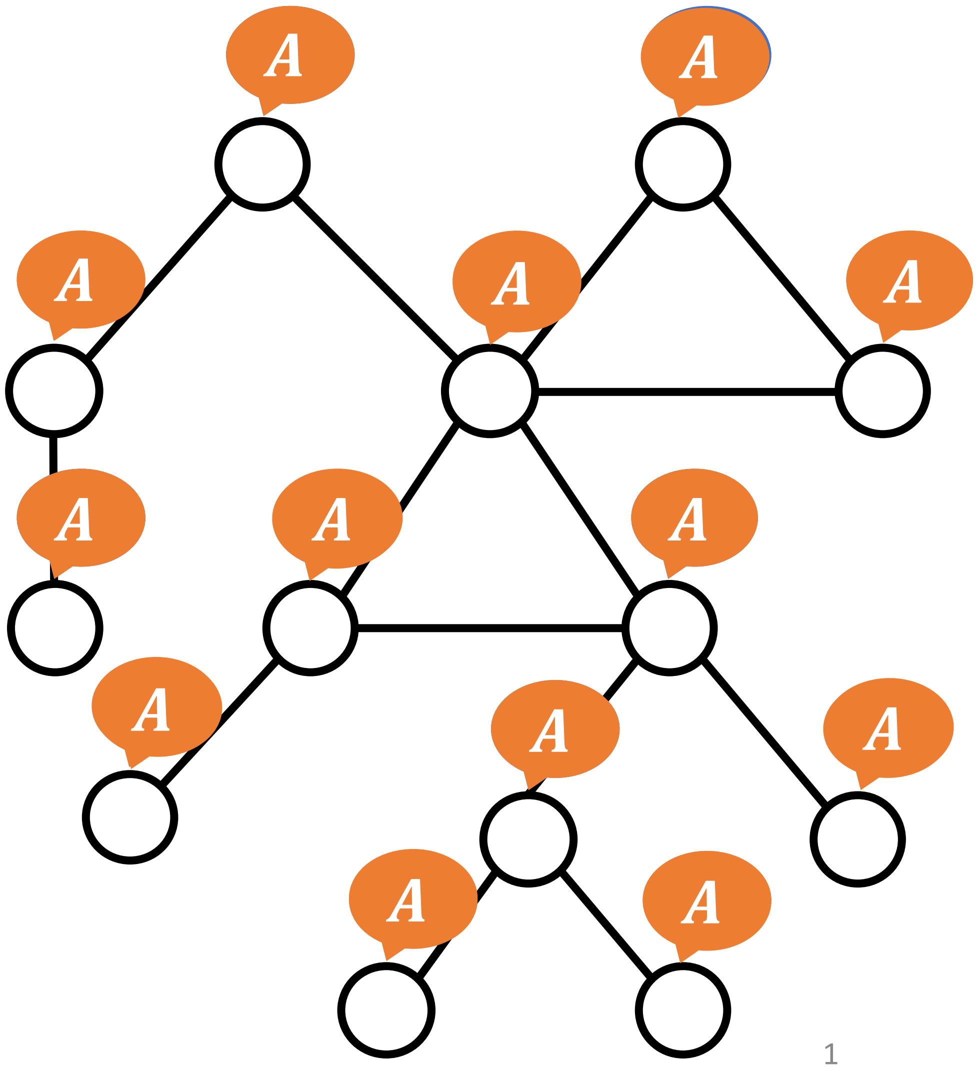
Population protocols & exact majority

- Distributed setting
  - Multiple agents (finite state automatons)
  - Each agent runs the same algorithm (protocol)
- Given:
  - Undirected graph G
  - Input opinions $\in \{A, B\}$



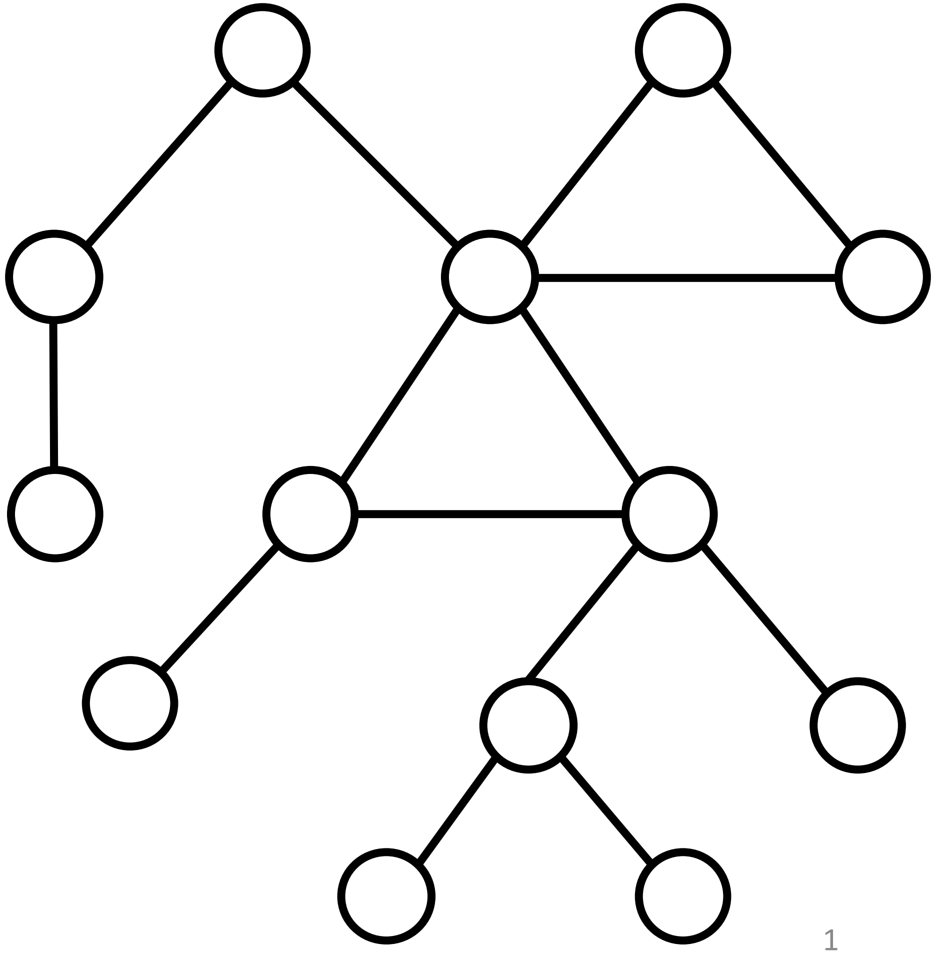
Population protocols & exact majority

- Distributed setting
  - Multiple agents (finite state automatons)
  - Each agent runs the same algorithm (protocol)
- Given:
  - Undirected graph G
  - Input opinions $\in \{A, B\}$
- Goal:
  - Every node outputs majority opinion a.s.



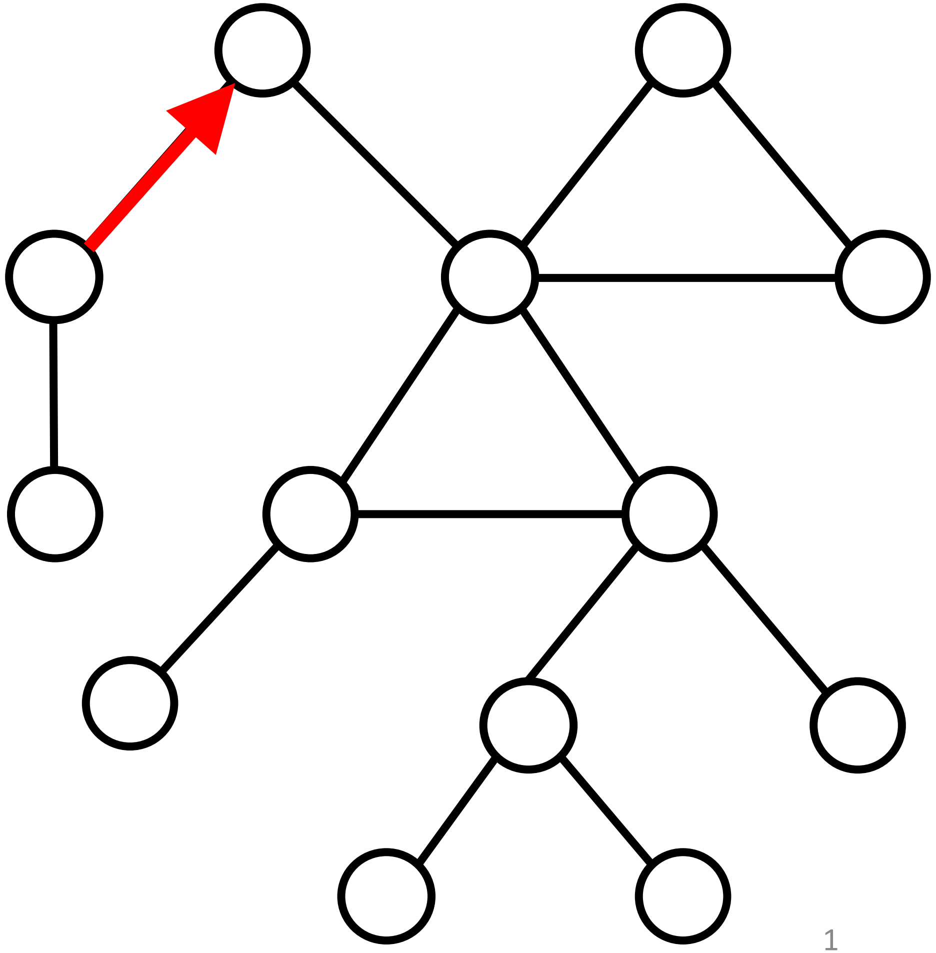
Population protocols & exact majority

- Distributed setting
  - Multiple agents (finite state automatons)
  - Each agent runs the same algorithm (protocol)
- Given:
  - Undirected graph G
  - Input opinions $\in \{A, B\}$
- Goal:
  - Every node outputs majority opinion a.s.
- Asynchronous communication steps:



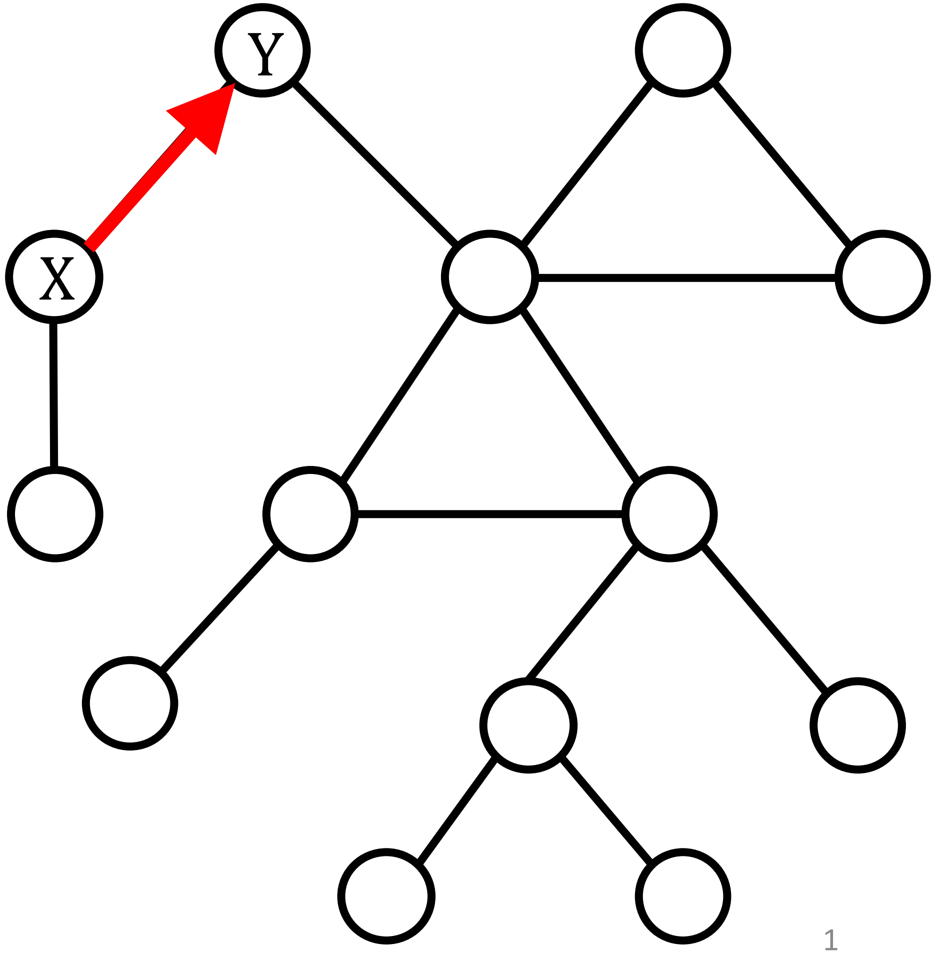
Population protocols & exact majority

- Distributed setting
  - Multiple agents (finite state automatons)
  - Each agent runs the same algorithm (protocol)
- Given:
  - Undirected graph G
  - Input opinions $\in \{A, B\}$
- Goal:
  - Every node outputs majority opinion a.s.
- Asynchronous communication steps:
  - Uniformly sample a directed edge (u, v)



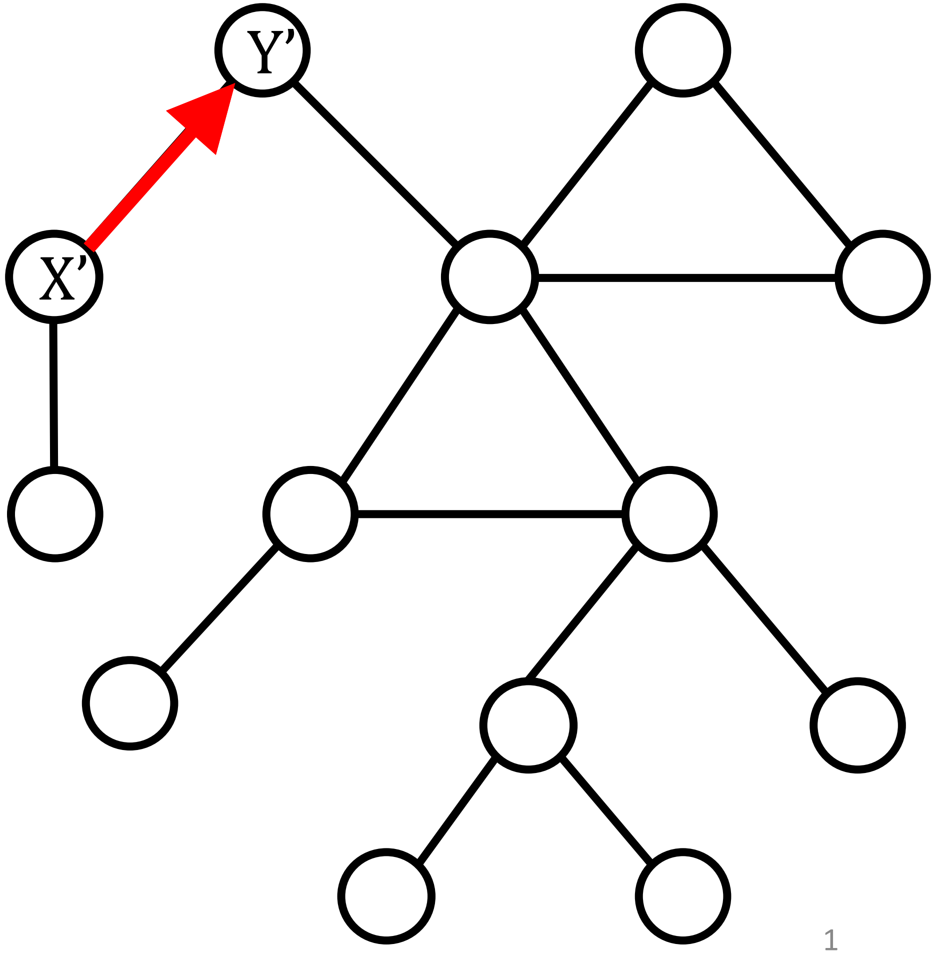
Population protocols & exact majority

- Distributed setting
  - Multiple agents (finite state automatons)
  - Each agent runs the same algorithm (protocol)
- Given:
  - Undirected graph G
  - Input opinions $\in \{A, B\}$
- Goal:
  - Every node outputs majority opinion a.s.
- Asynchronous communication steps:
  - Uniformly sample a directed edge (u, v)
  - Endpoints update their states: $\mathrm{X} + Y \to X' + Y'$



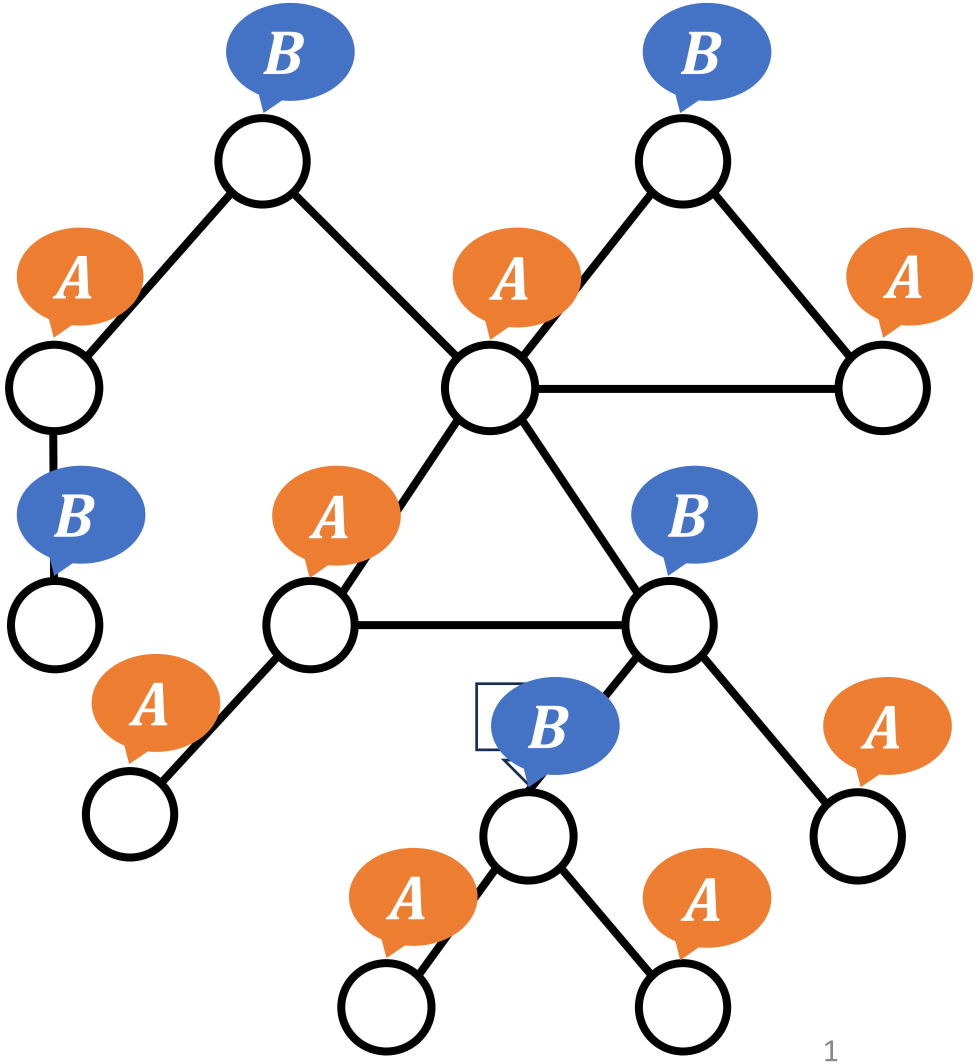
Population protocols & exact majority

- Distributed setting
  - Multiple agents (finite state automatons)
  - Each agent runs the same algorithm (protocol)
- Given:
  - Undirected graph G
  - Input opinions ∈ $\{A, B\}$
- Goal:
  - Every node outputs majority opinion a.s.
- Asynchronous communication steps:
  - Uniformly sample a directed edge (u, v)
  - Endpoints update their states: $X + Y \rightarrow X' + Y'$



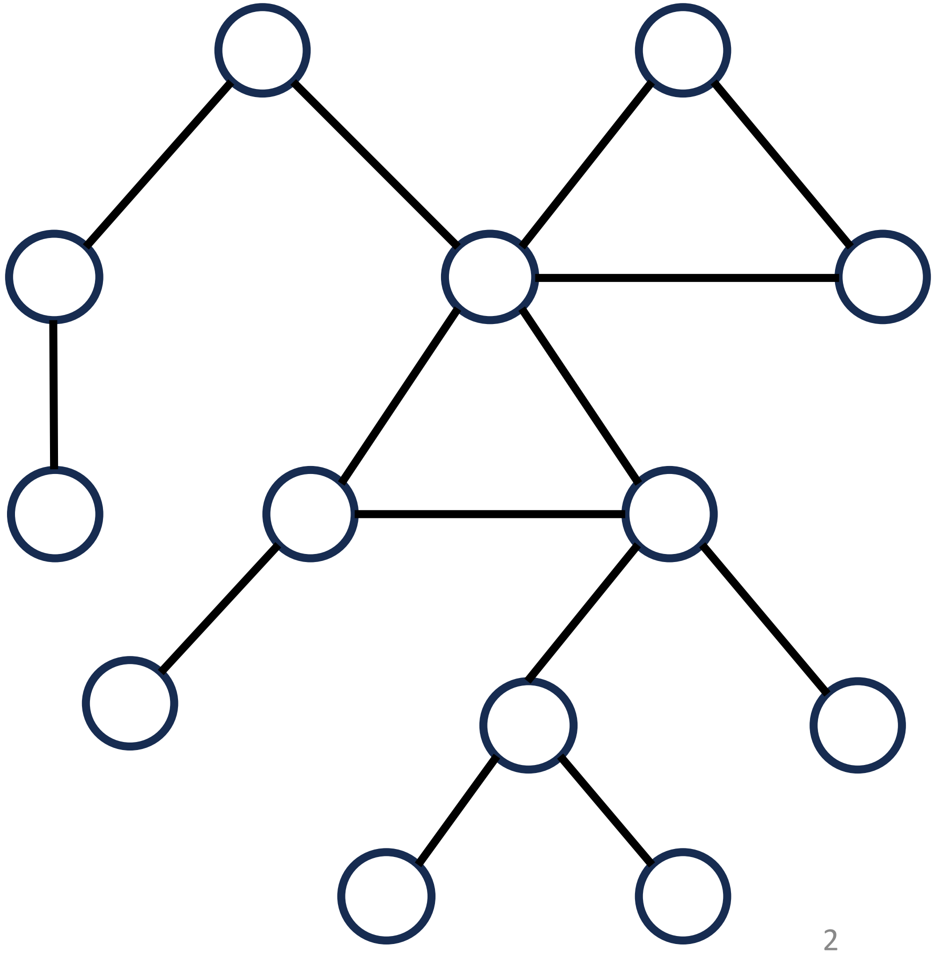
Population protocols & exact majority

- Distributed setting
  - Multiple agents (finite state automatons)
  - Each agent runs the same algorithm (protocol)
- Given:
  - Undirected graph G
  - Input opinions $\in \{A, B\}$
- Goal:
  - Every node outputs majority opinion a.s.
- Asynchronous communication steps:
  - Uniformly sample a directed edge (u, v)
  - Endpoints update their states: $X + Y \to X' + Y'$
  - Every node outputs depending on their state



[Bénézit et al., ICASSP’09]
[Draief and Vojnović, INFOCOM’10]
[Mertzios et al., ICALP’14]

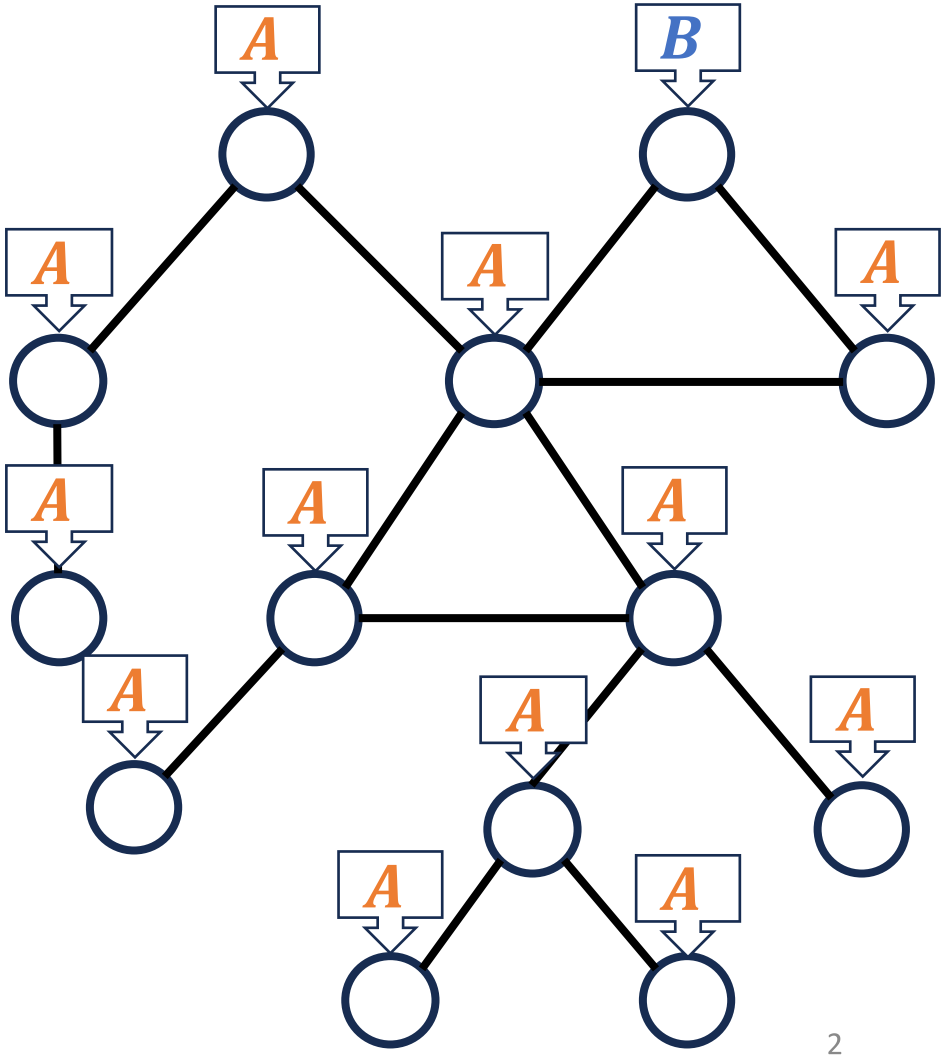
Example: a 4-state protocol



[Bénézit et al., ICASSP'09]
[Draief and Vojnović, INFOCOM'10]
[Mertzios et al., ICALP'14]

Example: a 4-state protocol

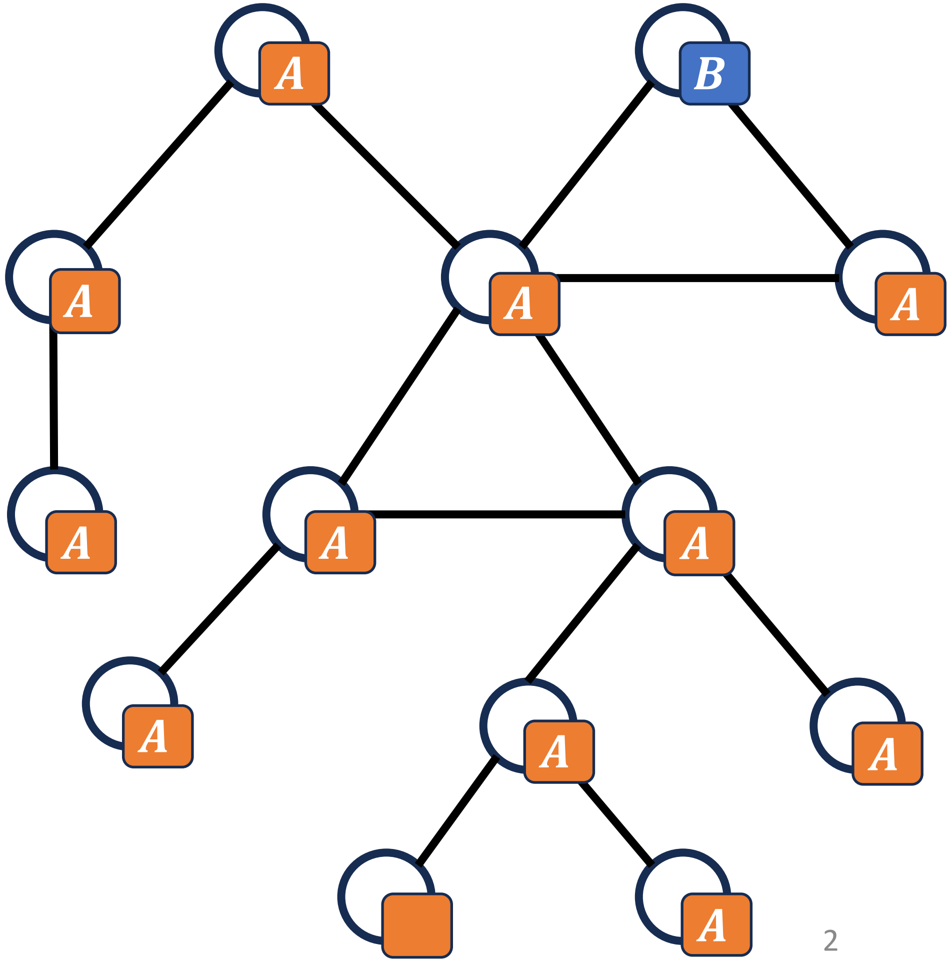
- States $= \{A, B, a, b\}$



[Bénézit et al., ICASSP'09]
[Draief and Vojnović, INFOCOM'10]
[Mertzios et al., ICALP'14]

Example: a 4-state protocol

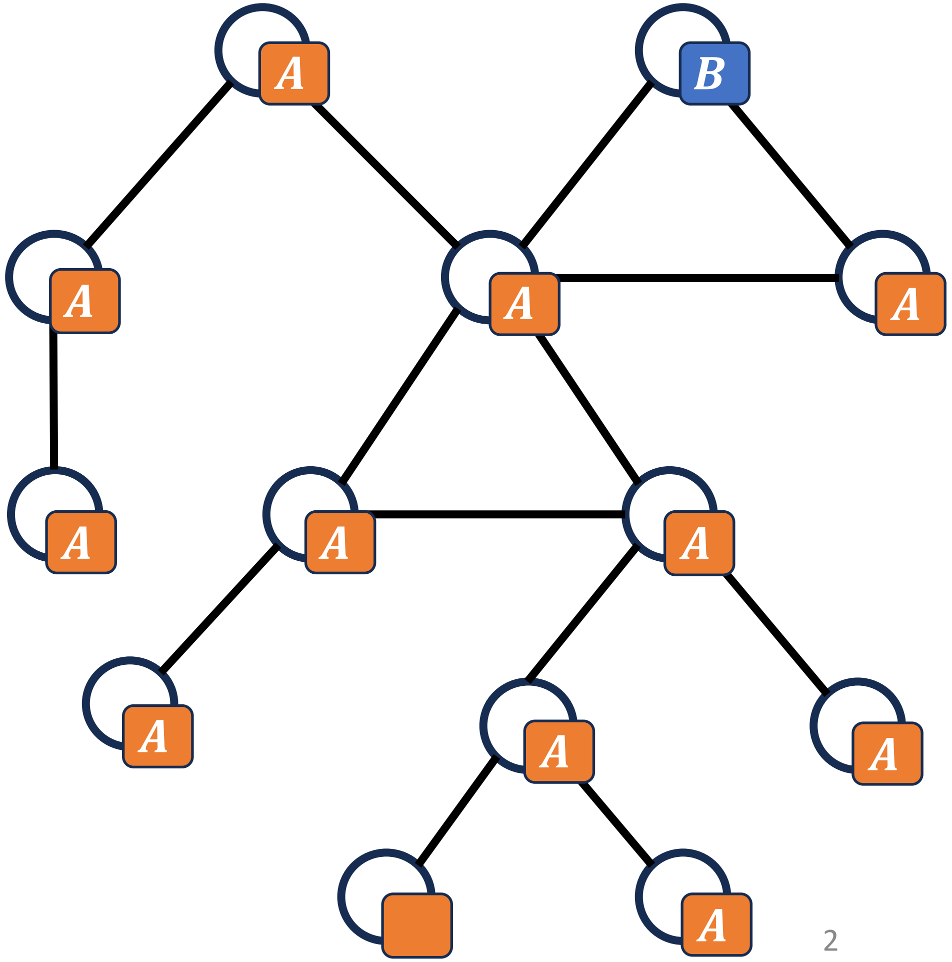
- States = $\{A, B, a, b\}$
- Initial state is input opinion



[Bénézit et al., ICASSP'09]
[Draief and Vojnović, INFOCOM'10]
[Mertzios et al., ICALP'14]

Example: a 4-state protocol

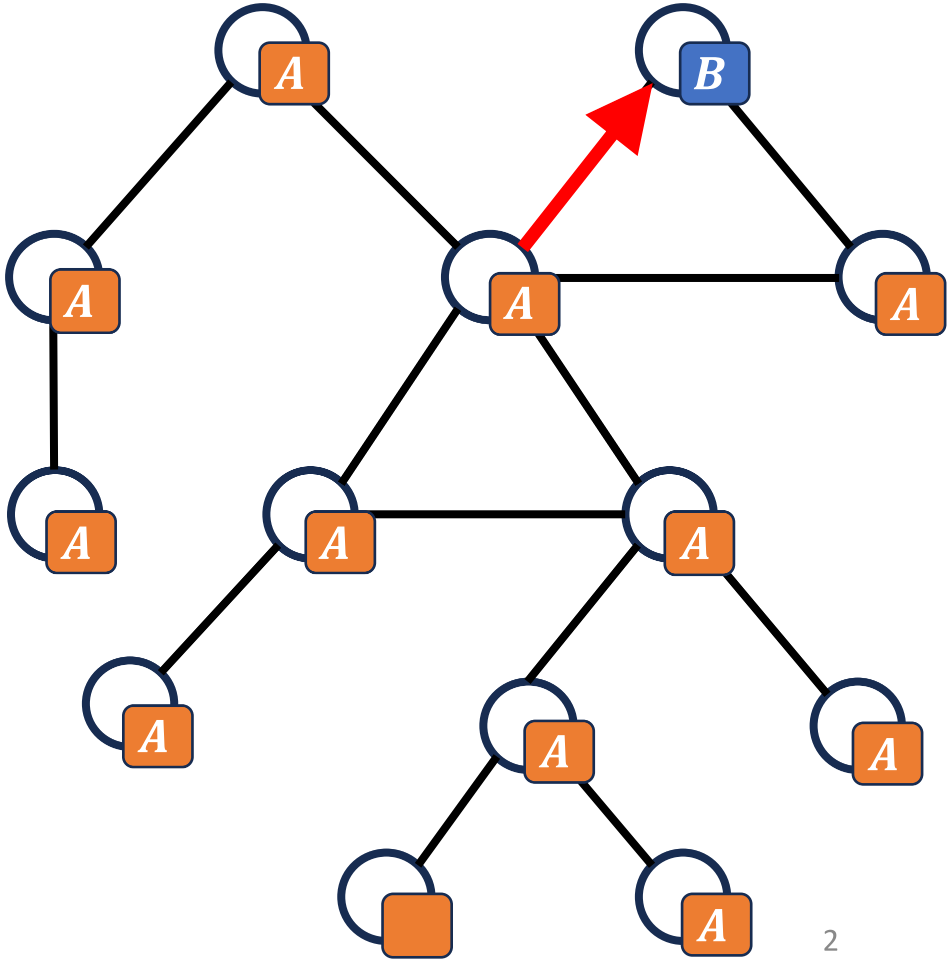
- States = $\{A, B, a, b\}$
- Initial state is input opinion
- Transition rules:



[Bénézit et al., ICASSP’09]
[Draief and Vojnović, INFOCOM’10]
[Mertzios et al., ICALP’14]

Example: a 4-state protocol

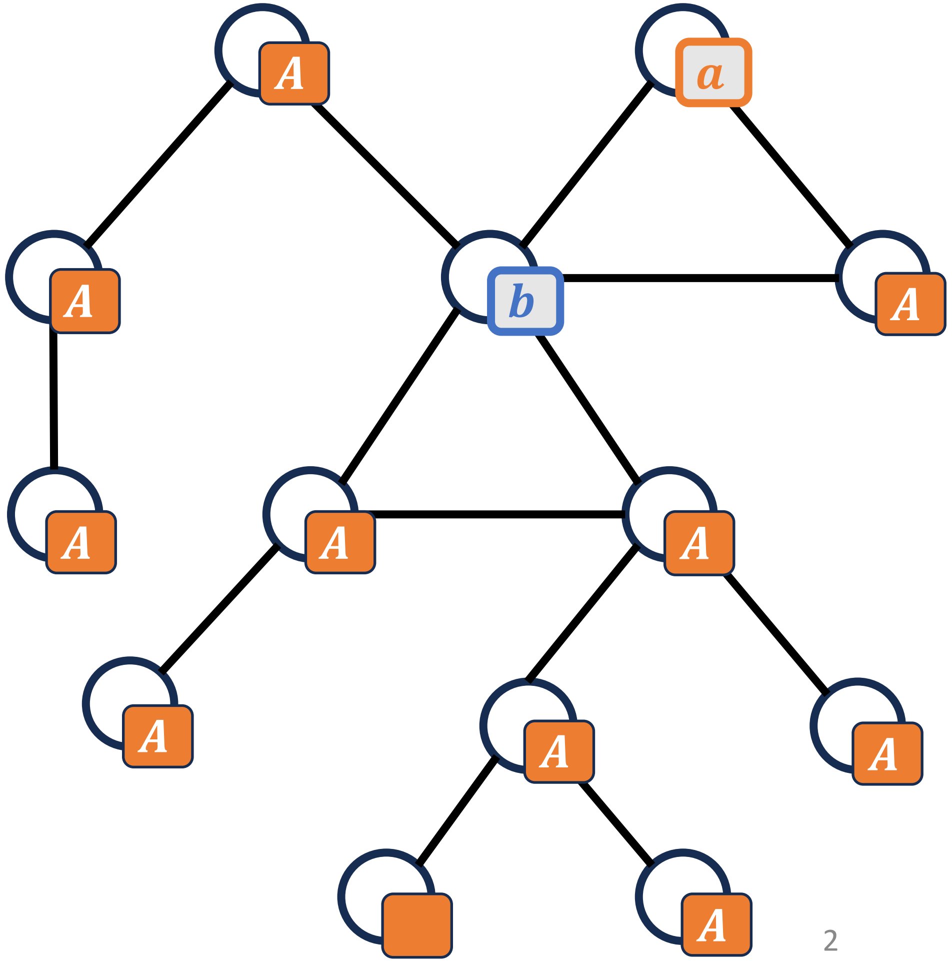
- States $= \{A, B, a, b\}$
- Initial state is input opinion
- Transition rules:
  - $A + B \rightarrow b + a$



[Bénézit et al., ICASSP'09]
[Draief and Vojnović, INFOCOM'10]
[Mertzios et al., ICALP'14]

Example: a 4-state protocol

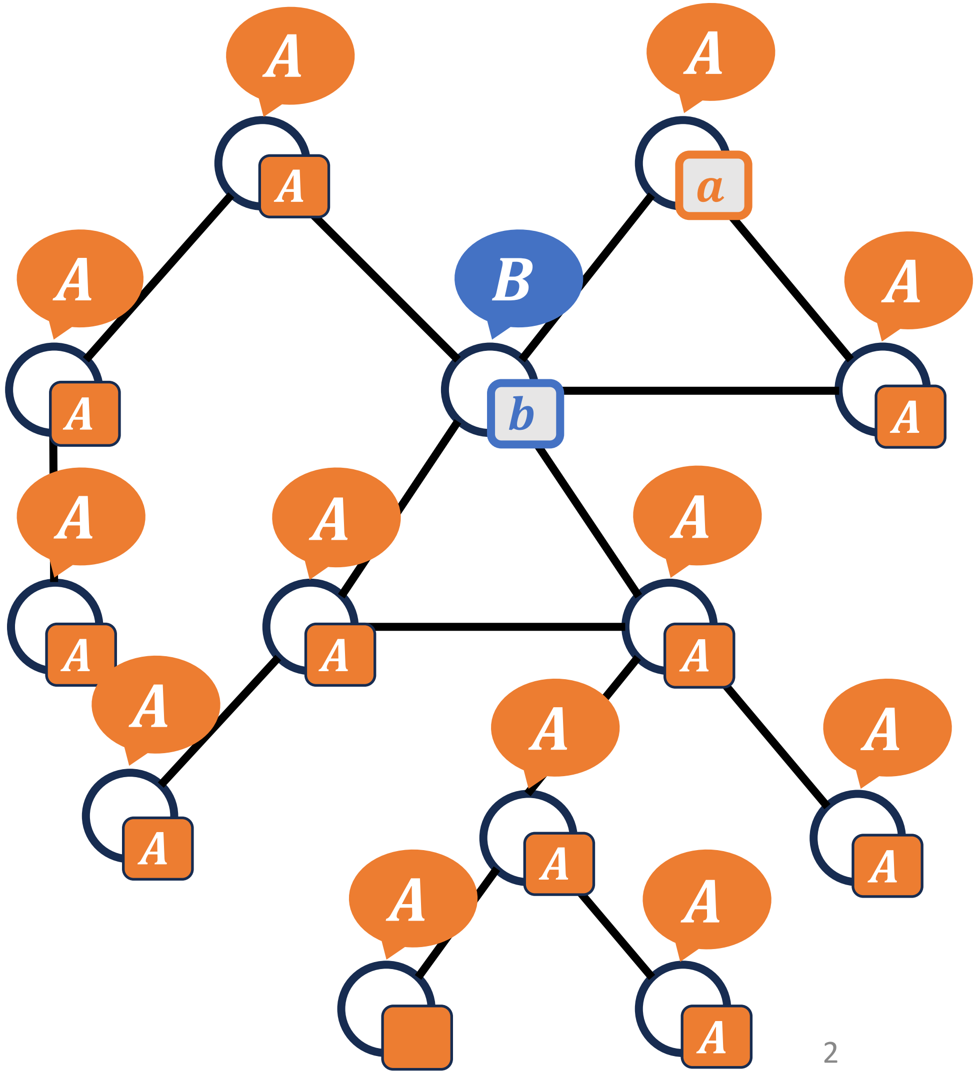
- States $= \{A, B, a, b\}$
- Initial state is input opinion
- Transition rules:
  - $A + B \rightarrow b + a$



[Bénézit et al., ICASSP’09]
[Draief and Vojnović, INFOCOM’10]
[Mertzios et al., ICALP’14]

Example: a 4-state protocol

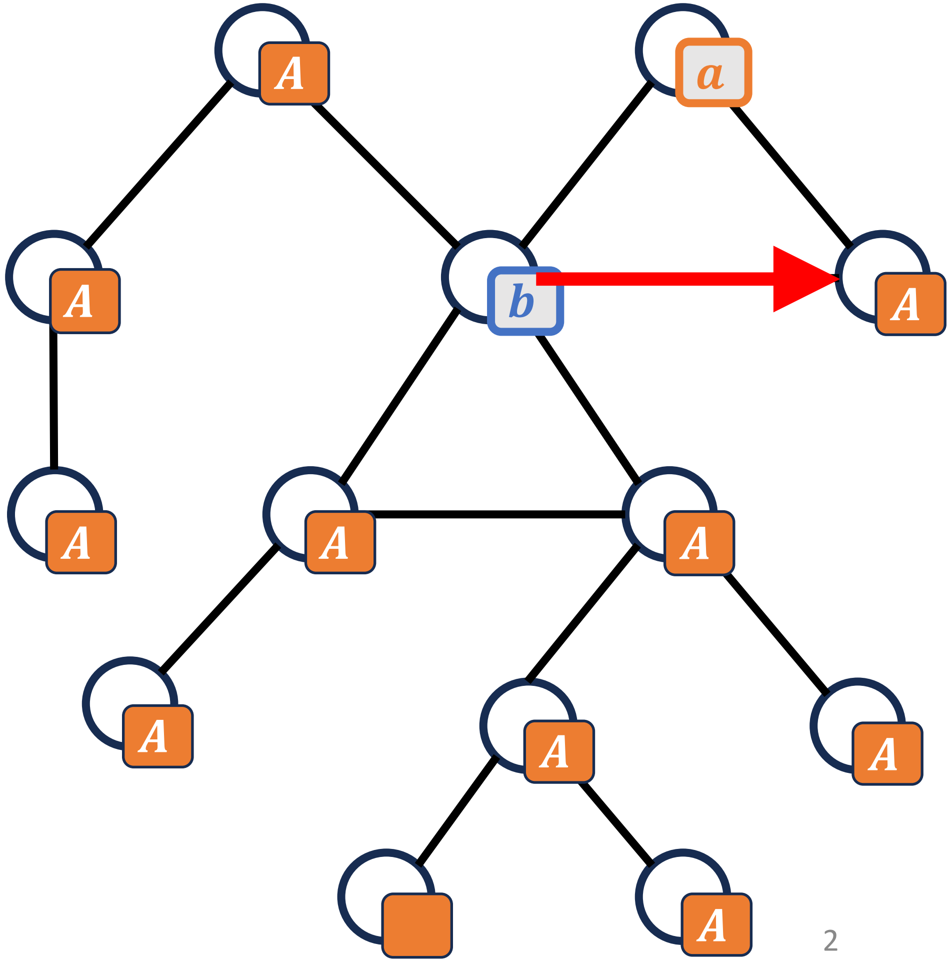
- States $= \{A, B, a, b\}$
- Initial state is input opinion
- Transition rules:
  - $A + B \rightarrow b + a$



[Bénézit et al., ICASSP’09]
[Draief and Vojnović, INFOCOM’10]
[Mertzios et al., ICALP’14]

Example: a 4-state protocol

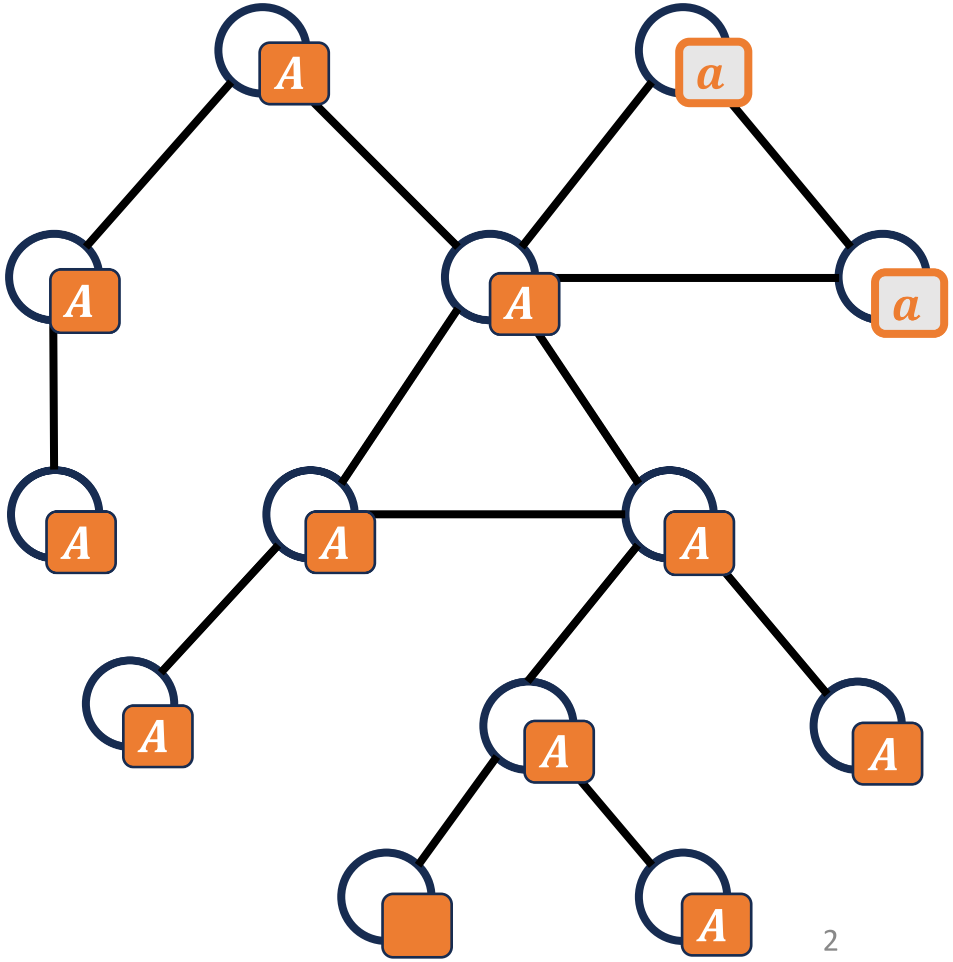
- States $= \{A, B, a, b\}$
- Initial state is input opinion
- Transition rules:
  - $A + B \rightarrow b + a$
  - $A + b \rightarrow a + A$ and $B + a \rightarrow b + B$



[Bénézit et al., ICASSP’09]
[Draief and Vojnović, INFOCOM’10]
[Mertzios et al., ICALP’14]

Example: a 4-state protocol

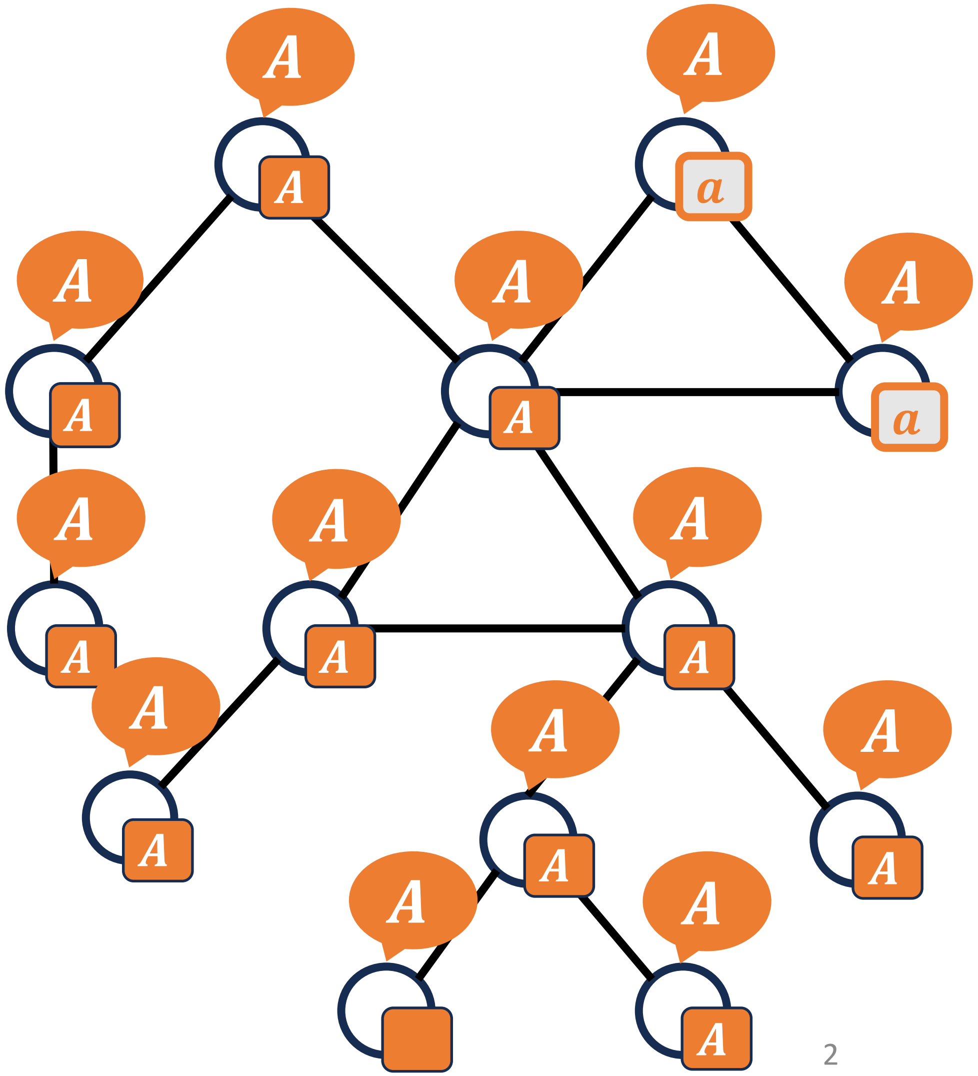
- States $= \{A, B, a, b\}$
- Initial state is input opinion
- Transition rules:
  - $A + B \rightarrow b + a$
  - $A + b \rightarrow a + A$ and $B + a \rightarrow b + B$



[Bénézit et al., ICASSP'09]
[Draief and Vojnović, INFOCOM'10]
[Mertzios et al., ICALP'14]

Example: a 4-state protocol

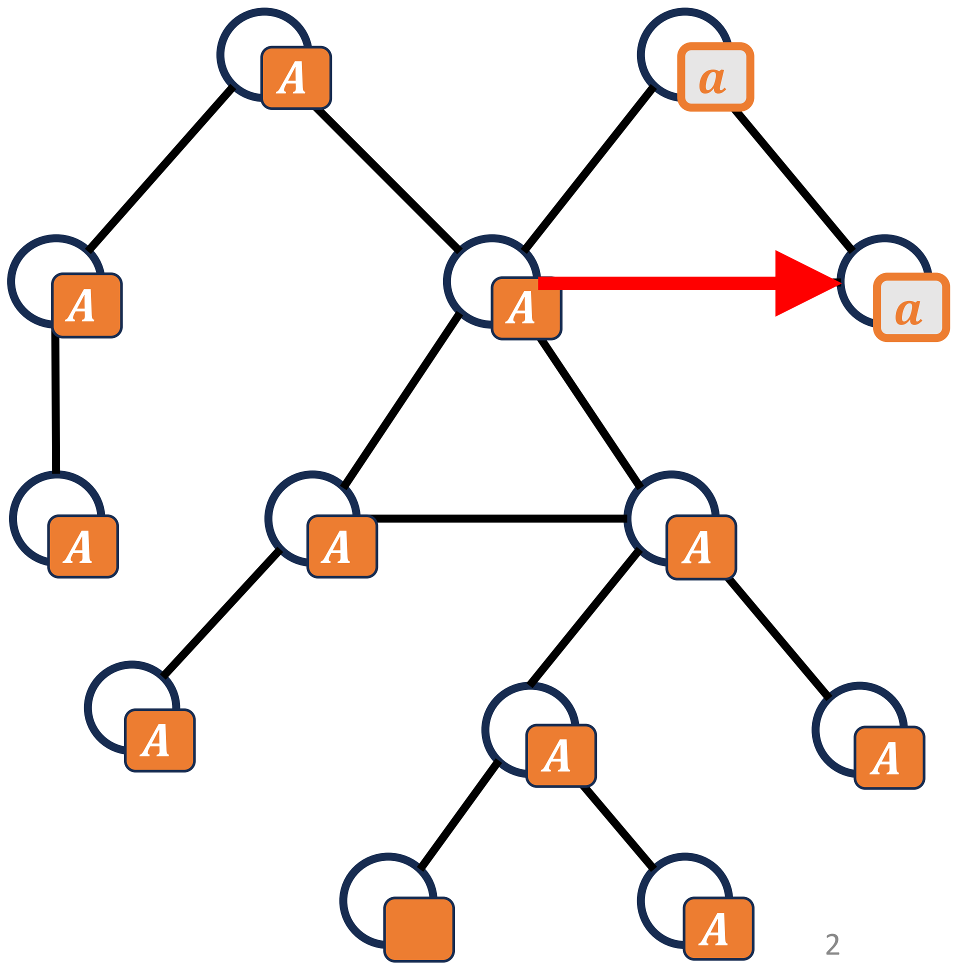
- States $= \{A, B, a, b\}$
- Initial state is input opinion
- Transition rules:
  - $A + B \to b + a$
  - $A + b \to a + A$ and $B + a \to b + B$



[Bénézit et al., ICASSP'09]
[Draief and Vojnović, INFOCOM'10]
[Mertzios et al., ICALP'14]

Example: a 4-state protocol

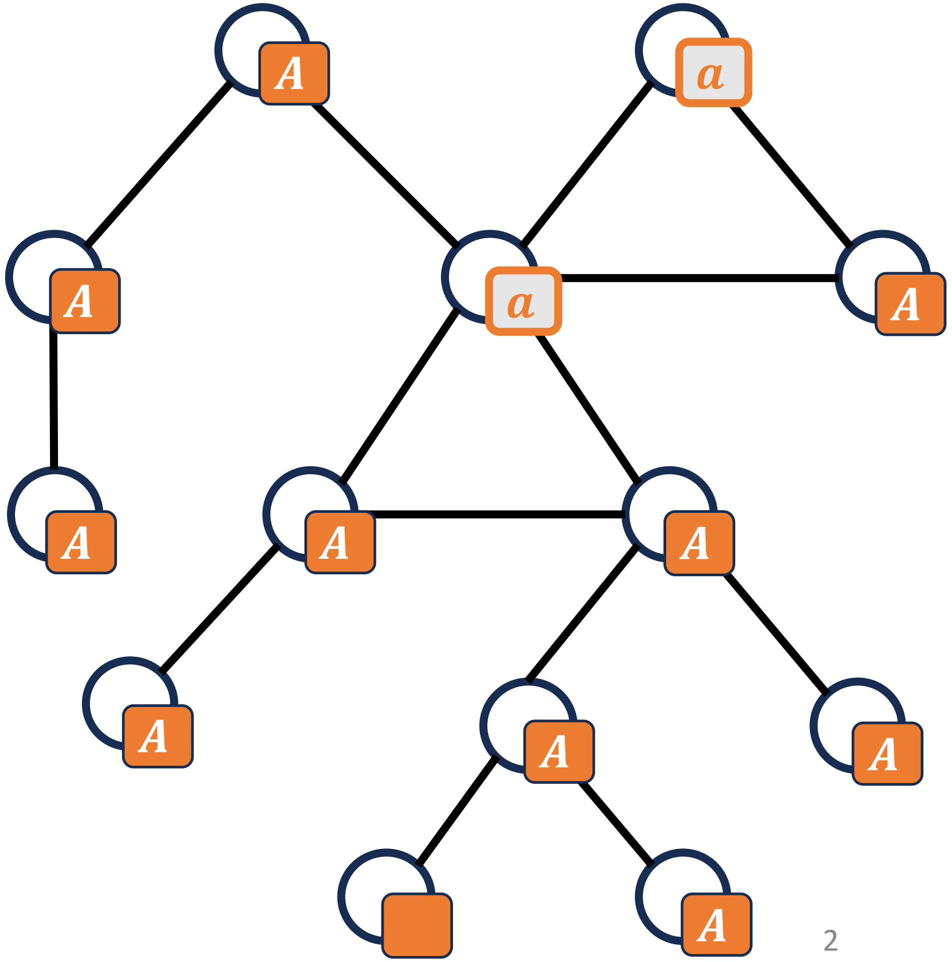
- States $= \{A, B, a, b\}$
- Initial state is input opinion
- Transition rules:
  - $A + B \rightarrow b + a$
  - $A + b \rightarrow a + A$ and $B + a \rightarrow b + B$
  - Otherwise swap: $X + Y \rightarrow Y + X$



[Bénézit et al., ICASSP'09]
[Draief and Vojnović, INFOCOM'10]
[Mertzios et al., ICALP'14]

Example: a 4-state protocol

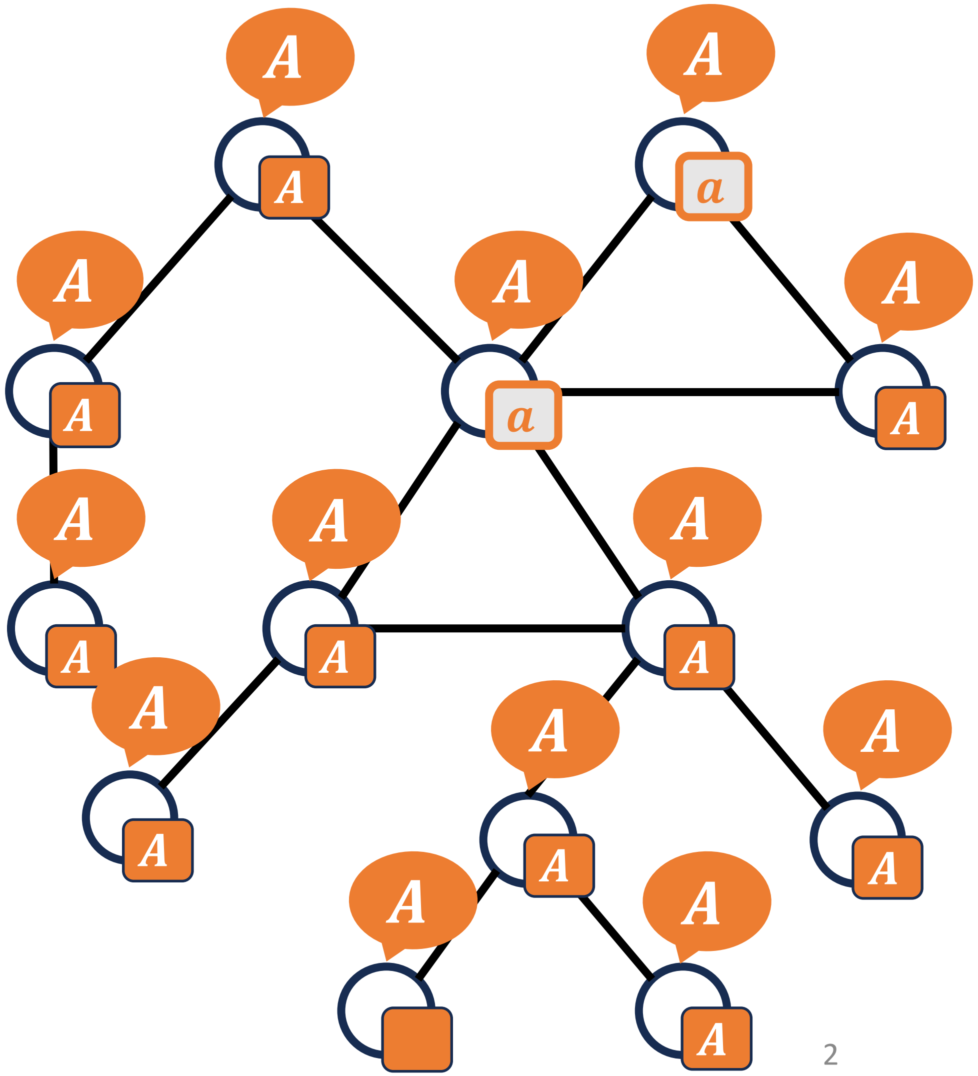
- States = $\{A, B, a, b\}$
- Initial state is input opinion
- Transition rules:
  - $A + B \rightarrow b + a$
  - $A + b \rightarrow a + A$ and $B + a \rightarrow b + B$
  - Otherwise swap: $X + Y \rightarrow Y + X$



[Bénézit et al., ICASSP’09]
[Draief and Vojnović, INFOCOM’10]
[Mertzios et al., ICALP’14]

Example: a 4-state protocol

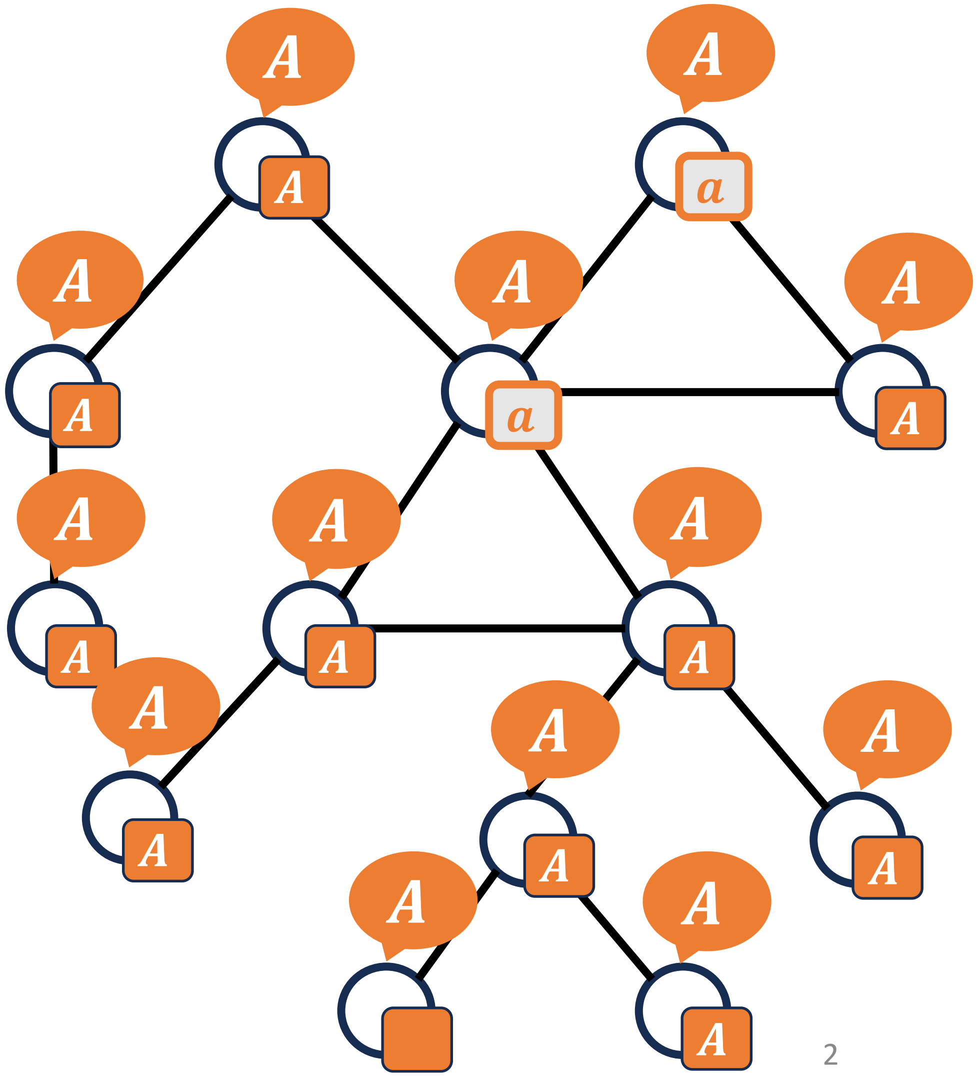
- States = $\{A, B, a, b\}$
- Initial state is input opinion
- Transition rules:
  - $A + B \rightarrow b + a$
  - $A + b \rightarrow a + A$ and $B + a \rightarrow b + B$
  - Otherwise swap: $X + Y \rightarrow Y + X$



Example: a 4-state protocol

[Bénézit et al., ICASSP'09]
[Draief and Vojnović, INFOCOM'10]
[Mertzios et al., ICALP'14]

- States $= \{A, B, a, b\}$
- Initial state is input opinion
- Transition rules:
  - $A + B \rightarrow b + a$
  - $A + b \rightarrow a + A$ and $B + a \rightarrow b + B$
  - Otherwise swap: $X + Y \rightarrow Y + X$
- Expected stabilization time (time complexity)

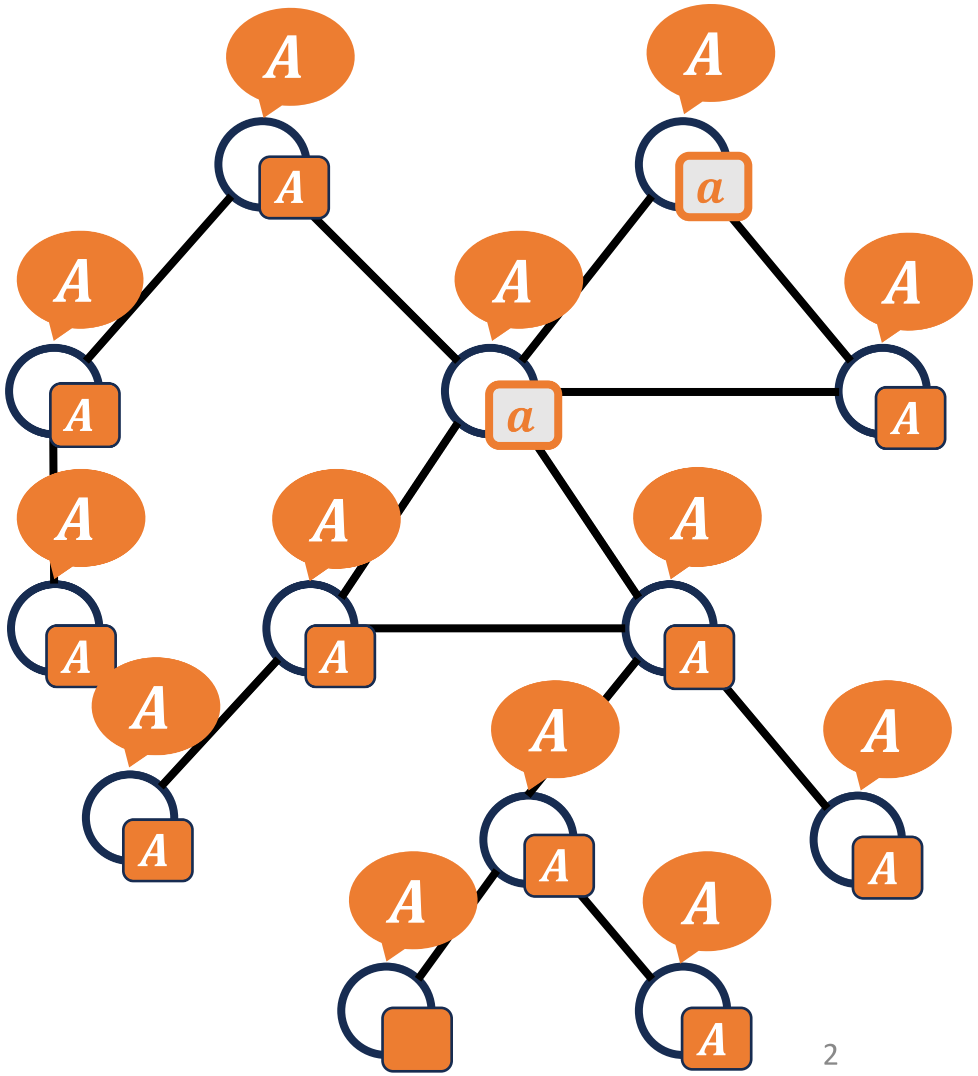


[Bénézit et al., ICASSP'09]
[Draief and Vojnović, INFOCOM'10]
[Mertzios et al., ICALP'14]

Example: a 4-state protocol

- States = $\{A, B, a, b\}$
- Initial state is input opinion
- Transition rules:
  - $A + B \to b + a$
  - $A + b \to a + A$ and $B + a \to b + B$
  - Otherwise swap: $X + Y \to Y + X$

- Expected stabilization time (time complexity)
- Number of states (space complexity)



State of the art for exact majority

State of the art for exact majority

- Well-mixed chemical solutions ≈ cliques

Graph	Stabilization time	Number of states	Reference
Clique	$O(n^2 \log n)$	4	Draief and Vojnović, INFOCOM’10
Clique	$O(n^{2-\varepsilon})$	$\Omega(\log n)$	Alistarh et al., SODA’18
Clique	$\Theta(n \log n)$	$\Theta(\log n)$	Doty et al., FOCS’22

State of the art for exact majority

- Well-mixed chemical solutions ≈ cliques

Graph	Stabilization time	Number of states	Reference
Clique	$O(n^2 \log n)$	4	Draief and Vojnović, INFOCOM'10
Clique	$O(n^{2-\varepsilon})$	$\Omega(\log n)$	Alistarh et al., SODA'18
Clique	$\Theta(n \log n)$	$\Theta(\log n)$	Doty et al., FOCS'22

- Spatially structured systems ≈ general graphs

Connected	$O(n^6)$	4	Mertzios et al., ICALP'14
Connected	$O(\log n / \lambda(\gamma))$ (*)	4	Draief and Vojnović, INFOCOM'10
Regular	$O(\phi^{-2} n \log^6 n)$ (**)	$O(\phi^{-2} \log^5 n)$	Alistarh et al., OPODIS'21

(*) $\lambda(\gamma)$ = minimum spectral gap of certain family of submatrices

(**) ϕ = conductance

State of the art for exact majority

- Well-mixed chemical solutions ≈ cliques

Graph	Stabilization time	Number of states	Reference
Clique	$O(n^2 \log n)$	4	Draief and Vojnović, INFOCOM’10
Clique	$O(n^{2-\varepsilon})$	$\Omega(\log n)$	Alistarh et al., SODA’18
Clique	$\Theta(n \log n)$	$\Theta(\log n)$	Doty et al., FOCS’22

- Spatially structured systems ≈ general graphs

Connected	$O(n^6)$	4	Mertzios et al., ICALP’14
Connected	$O(\log n / \lambda(\gamma))$ (*)	4	Draief and Vojnović, INFOCOM’10
Regular	$O(\phi^{-2} n \log^6 n)$ (**)	$O(\phi^{-2} \log^5 n)$	Alistarh et al., OPODIS’21

⇒ Our focus: fast and space-efficient protocol for general graphs

(*) $\lambda(\gamma)$ = minimum spectral gap of certain family of submatrices

(**) ϕ = conductance

Cancellation & doubling

e.g. [Angluin et al., DISC'06]
[Berenbrink et al., DISC'18]
[Alistarh et al., OPODIS'21]
[Doty et al., FOCS'22]

Cancellation & doubling

- All known fast protocols for exact majority rely on this framework

e.g. [Angluin et al., DISC’06]
[Berenbrink et al., DISC’18]
[Alistarh et al., OPODIS’21]
[Doty et al., FOCS’22]

Cancellation & doubling

- All known fast protocols for exact majority rely on this framework
- Initially each node creates an opinion token

e.g. [Angluin et al., DISC'06]
[Berenbrink et al., DISC'18]
[Alistarh et al., OPODIS'21]
[Doty et al., FOCS'22]

Cancellation & doubling

- All known fast protocols for exact majority rely on this framework
- Initially each node creates an opinion token
- Alternate two phases:

e.g. [Angluin et al., DISC’06]
[Berenbrink et al., DISC’18]
[Alistarh et al., OPODIS’21]
[Doty et al., FOCS’22]

Cancellation & doubling

- All known fast protocols for exact majority rely on this framework
- Initially each node creates an opinion token
- Alternate two phases:
  - Cancellation phase: $A + B \to C + C$

e.g. [Angluin et al., DISC'06]
[Berenbrink et al., DISC'18]
[Alistarh et al., OPODIS'21]
[Doty et al., FOCS'22]

Cancellation & doubling

- All known fast protocols for exact majority rely on this framework
- Initially each node creates an opinion token
- Alternate two phases:
  - Cancellation phase: $A + B \rightarrow C + C$
  - Doubling phase: $A + C \rightarrow a + a$
    $B + C \rightarrow b + b$

e.g. [Angluin et al., DISC'06]
[Berenbrink et al., DISC'18]
[Alistarh et al., OPODIS'21]
[Doty et al., FOCS'22]

Cancellation & doubling

- All known fast protocols for exact majority rely on this framework
- Initially each node creates an opinion token
- Alternate two phases:
  - Cancellation phase: $A + B \to C + C$
  - Doubling phase: $A + C \to a + a$, $B + C \to b + b$

Otherwise swap:
$X + Y \to Y + X$

e.g. [Angluin et al., DISC'06]
[Berenbrink et al., DISC'18]
[Alistarh et al., OPODIS'21]
[Doty et al., FOCS'22]

Cancellation & doubling

- All known fast protocols for exact majority rely on this framework
- Initially each node creates an opinion token
- Alternate two phases:
  - Cancellation phase: $A + B \to C + C$
  - Doubling phase: $A + C \to a + a$, $B + C \to b + b$

Otherwise swap:
$X + Y \to Y + X$

- Two key challenges:

e.g. [Angluin et al., DISC'06]
[Berenbrink et al., DISC'18]
[Alistarh et al., OPODIS'21]
[Doty et al., FOCS'22]

Cancellation & doubling

- All known fast protocols for exact majority rely on this framework
- Initially each node creates an opinion token
- Alternate two phases:
  - Cancellation phase: $A + B \rightarrow C + C$
  - Doubling phase: $A + C \rightarrow a + a$, $B + C \rightarrow b + b$

Otherwise swap:
$X + Y \rightarrow Y + X$

- Two key challenges:
  - 1) How to synchronize nodes in this asynchronous model? ($\rightarrow$ phase clock)

Cancellation & doubling

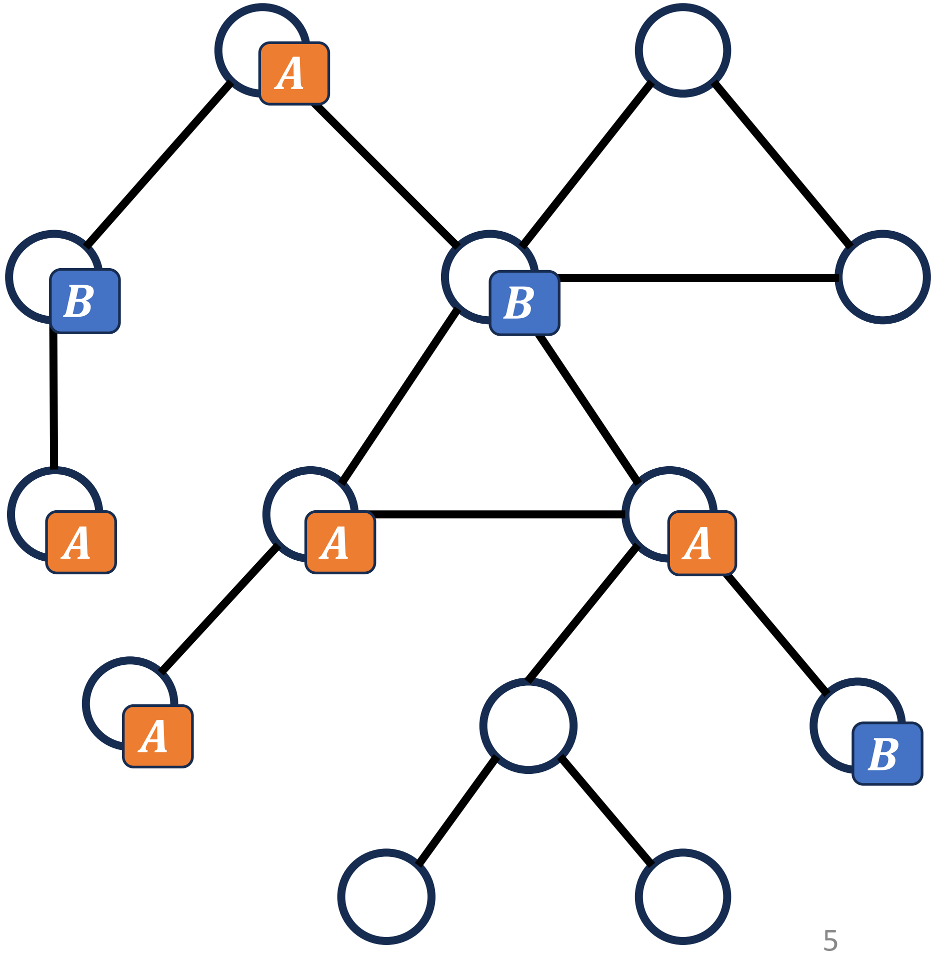
e.g. [Angluin et al., DISC'06]
[Berenbrink et al., DISC'18]
[Alistarh et al., OPODIS'21]
[Doty et al., FOCS'22]

- All known fast protocols for exact majority rely on this framework
- Initially each node creates an opinion token
- Alternate two phases:
  - Cancellation phase: $A + B \to C + C$
  - Doubling phase: $A + C \to a + a$, $B + C \to b + b$

Otherwise swap:
$X + Y \to Y + X$

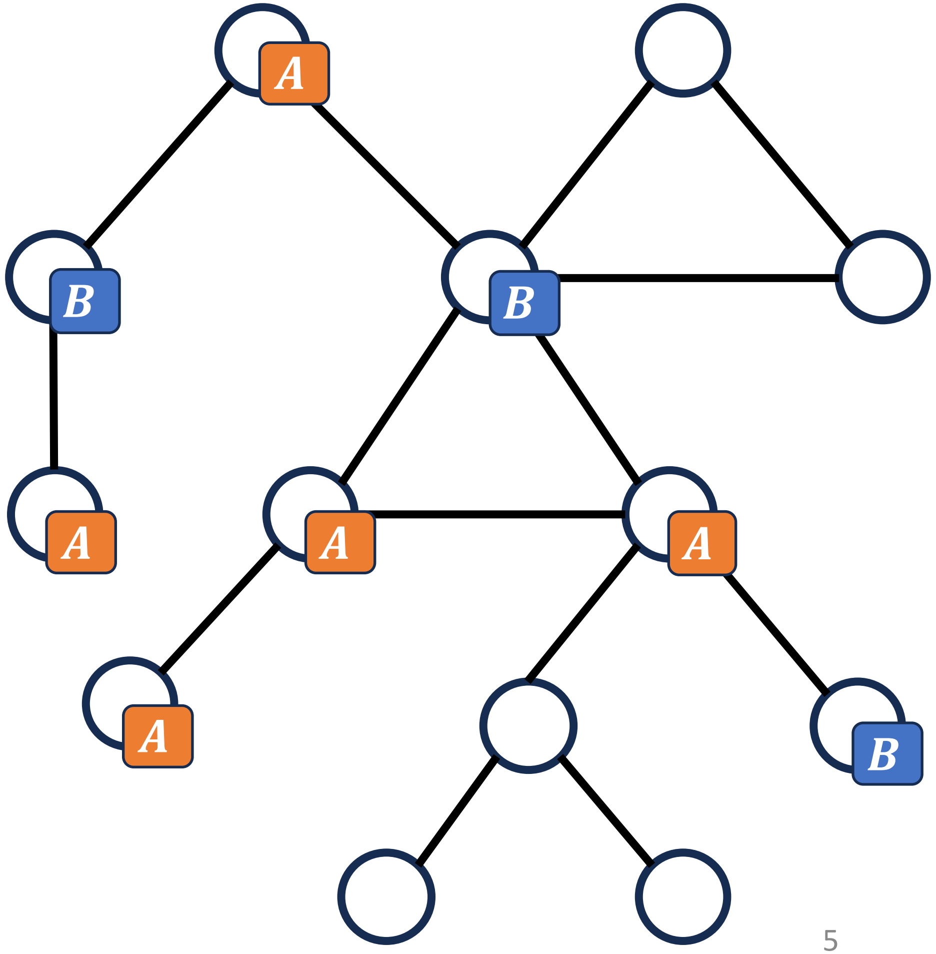
- Two key challenges:
  - 1) How to synchronize nodes in this asynchronous model? ($\to$ phase clock)
  - 2) How long to run each phase? ($\to$ annihilation dynamics)

Key ingredient: annihilation dynamics



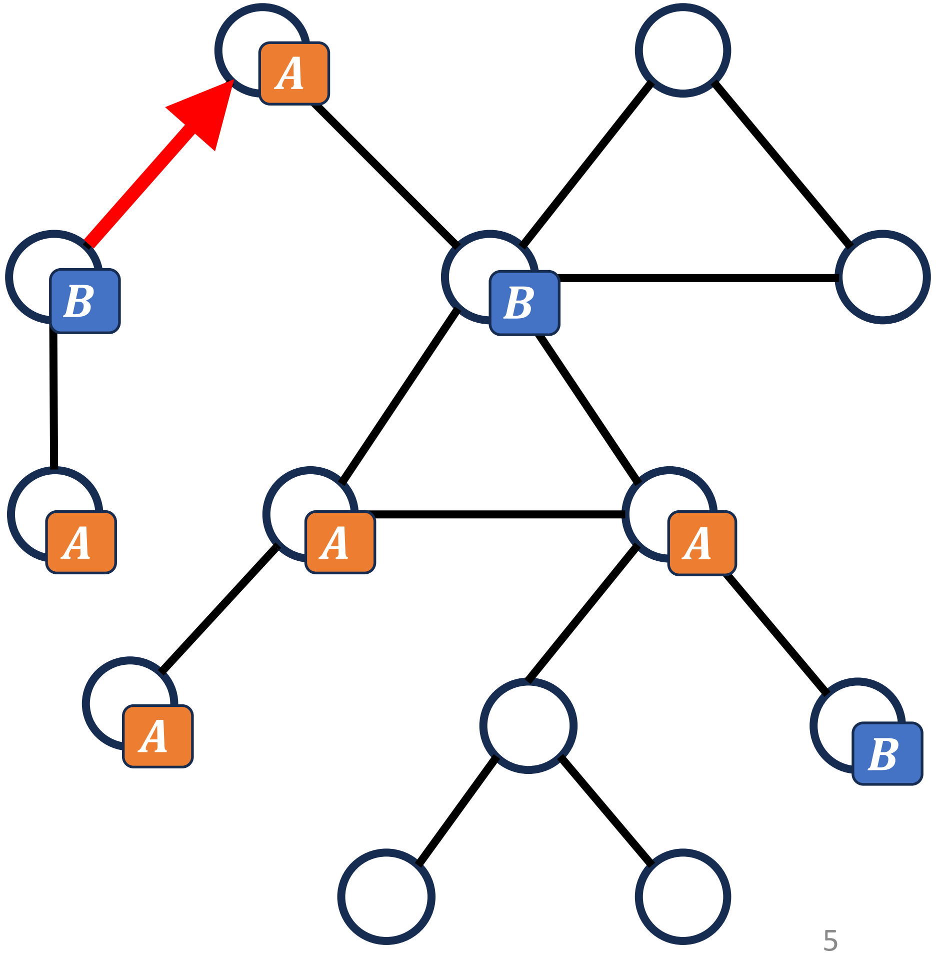
Key ingredient: annihilation dynamics

- Annihilation dynamics:



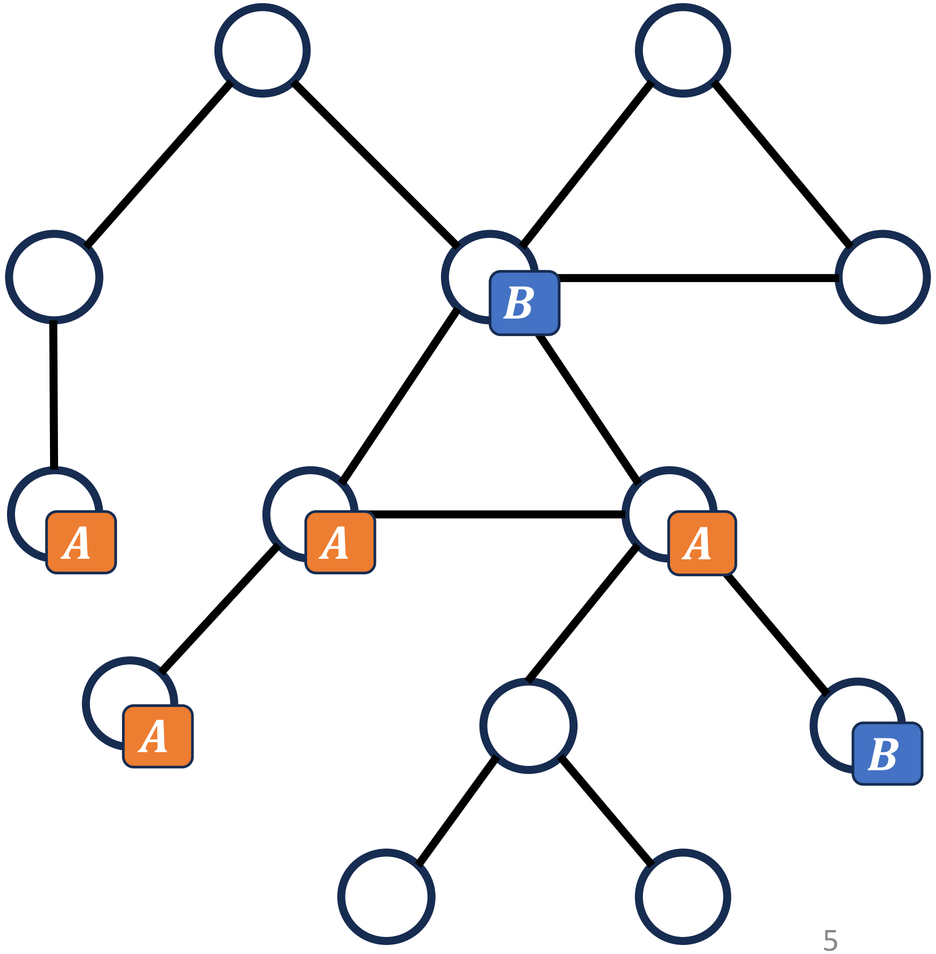
Key ingredient: annihilation dynamics

- Annihilation dynamics:
  - A and B tokens meet → annihilation



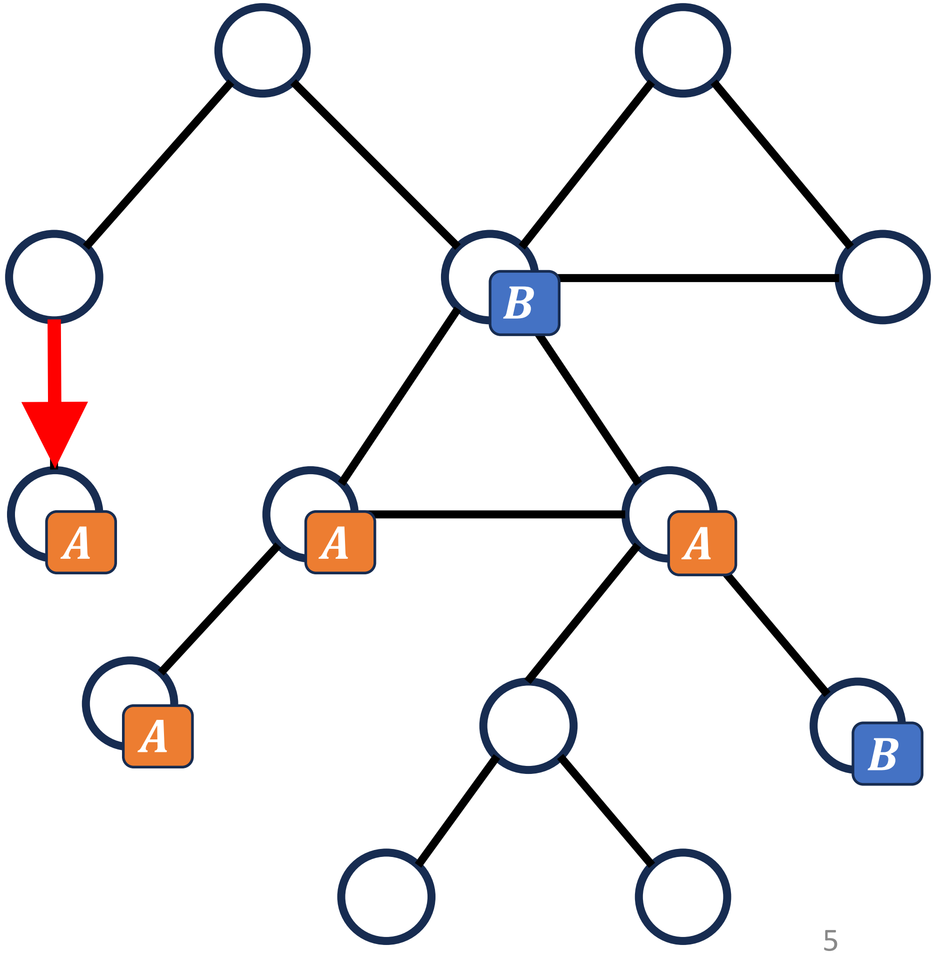
Key ingredient: annihilation dynamics

- Annihilation dynamics:
  - A and B tokens meet → annihilation



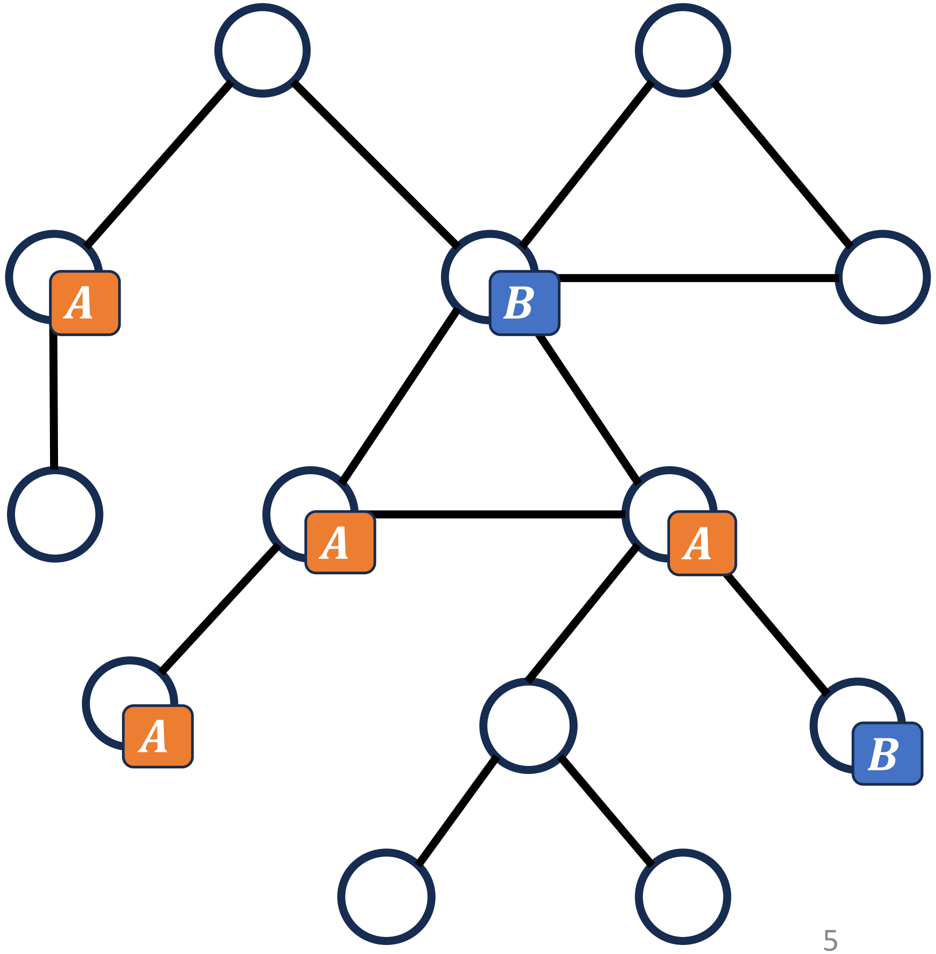
Key ingredient: annihilation dynamics

- Annihilation dynamics:
  - A and B tokens meet → annihilation
  - Incident edge of token sampled → token moves



Key ingredient: annihilation dynamics

- Annihilation dynamics:
  - A and B tokens meet → annihilation
  - Incident edge of token sampled → token moves

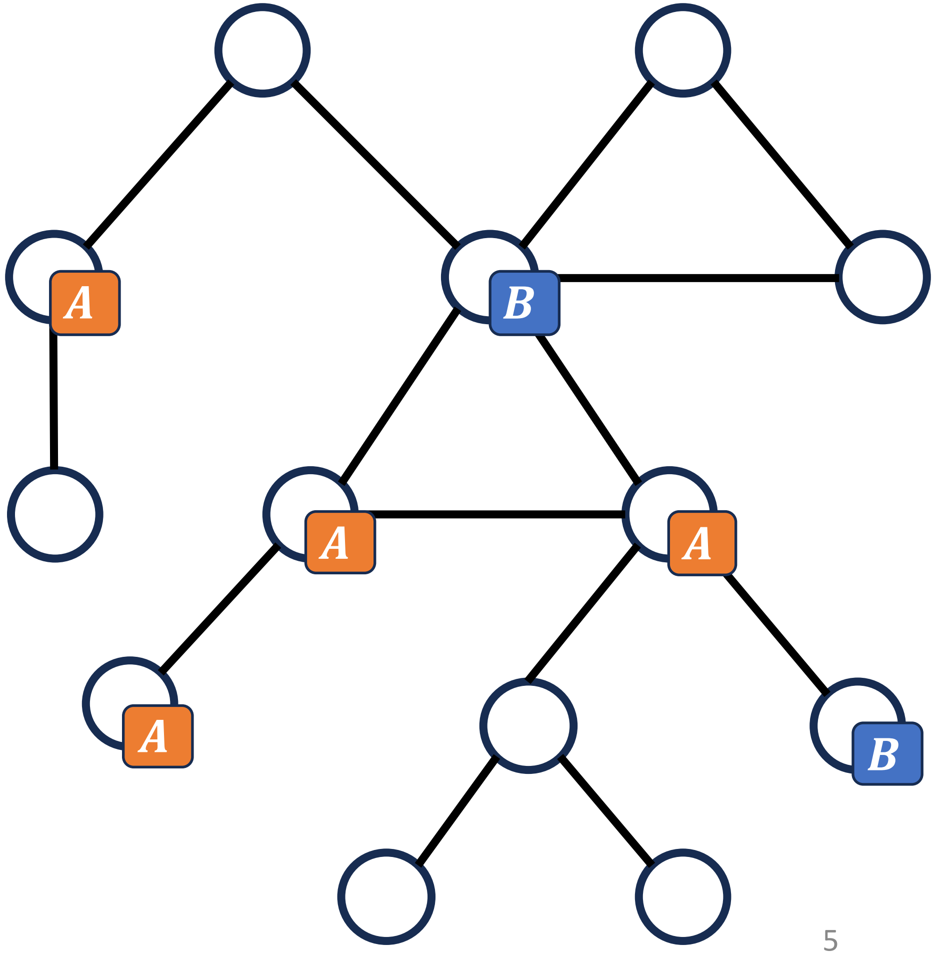


Key ingredient: annihilation dynamics

- Annihilation dynamics:
  - A and B tokens meet → annihilation
  - Incident edge of token sampled → token moves

- ε-clearing time

  (ε-fraction are A and B tokens or no minority)



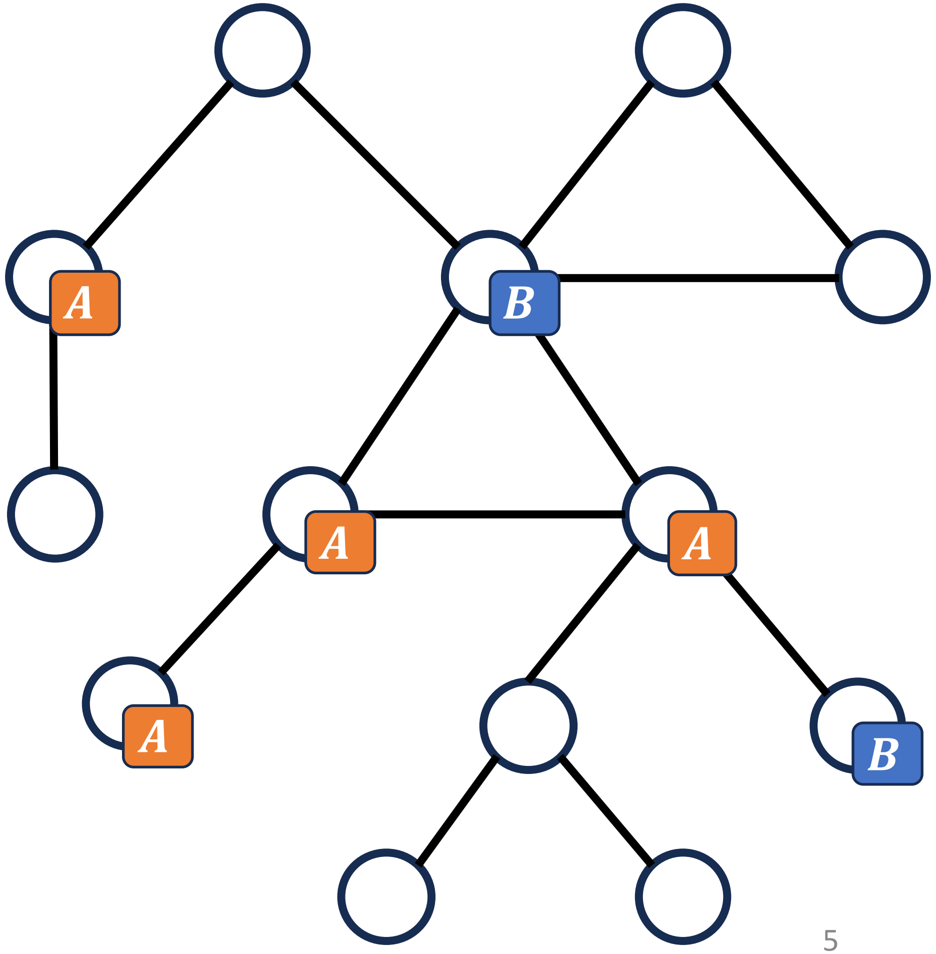
Key ingredient: annihilation dynamics

- Annihilation dynamics:
  - A and B tokens meet → annihilation
  - Incident edge of token sampled → token moves

- ε-clearing time

  (ε-fraction are A and B tokens or no minority)

  ↔ cancellation phase



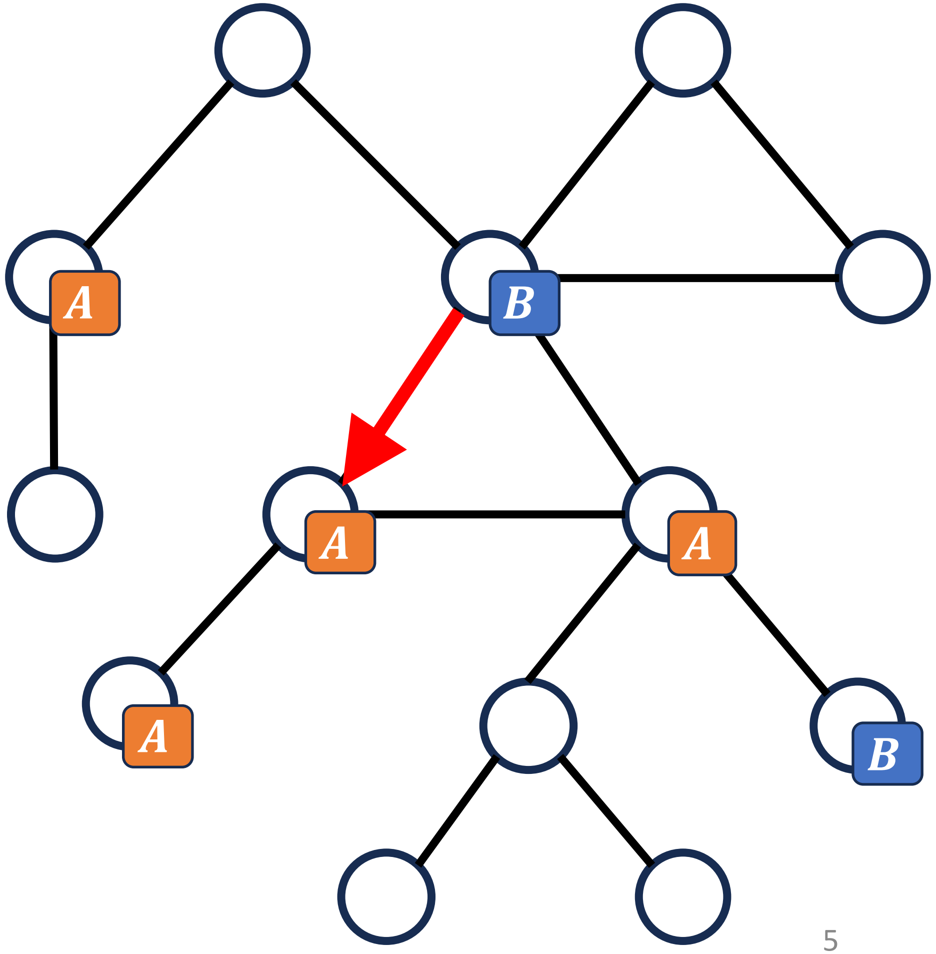
Key ingredient: annihilation dynamics

- Annihilation dynamics:
  - A and B tokens meet → annihilation
  - Incident edge of token sampled → token moves

- ε-clearing time

  (ε-fraction are A and B tokens or no minority)

  ↔ cancellation phase



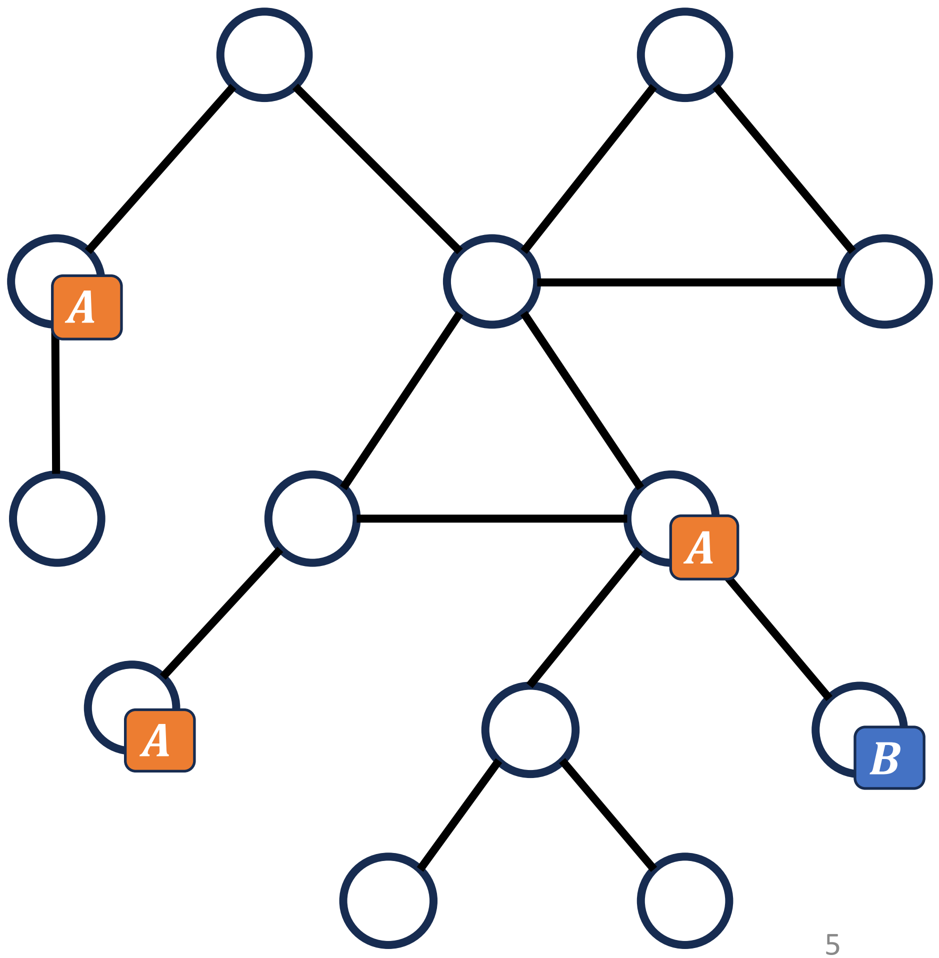
Key ingredient: annihilation dynamics

- Annihilation dynamics:
  - A and B tokens meet → annihilation
  - Incident edge of token sampled → token moves

- ε-clearing time

  (ε-fraction are A and B tokens or no minority)

  ↔ cancellation phase

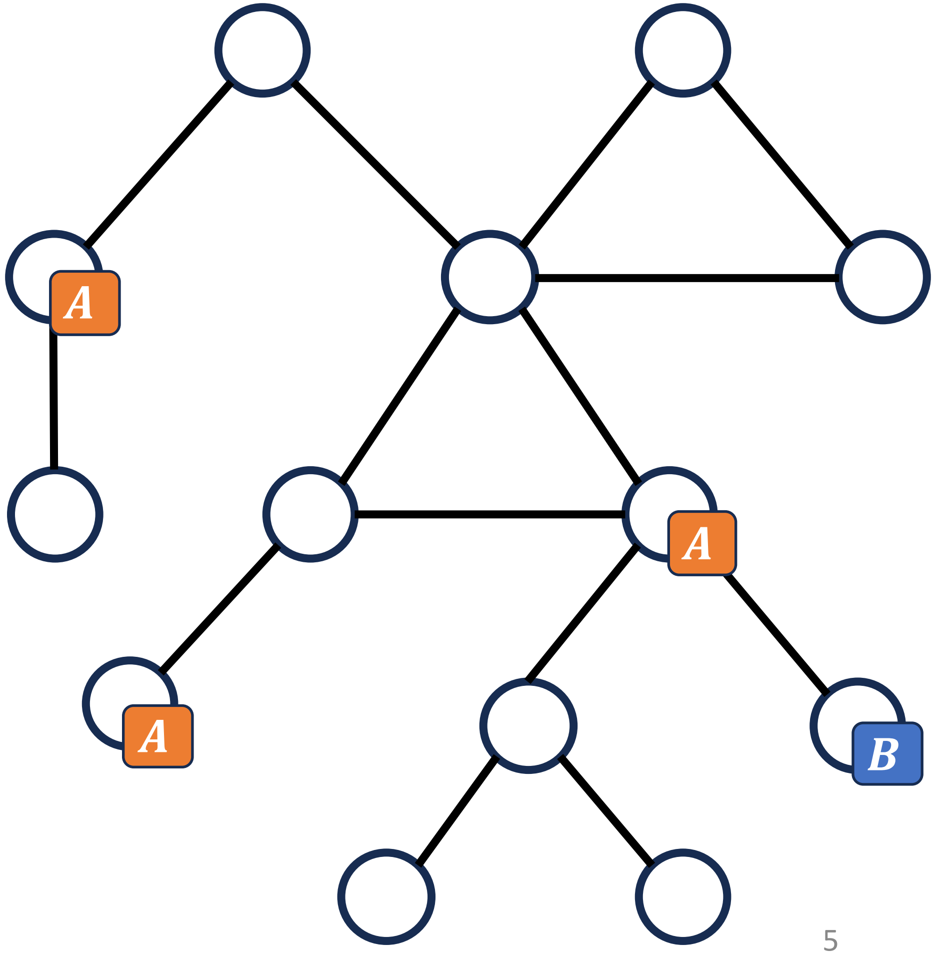


Key ingredient: annihilation dynamics

- Annihilation dynamics:
  - A and B tokens meet → annihilation
  - Incident edge of token sampled → token moves

- ε-clearing time
  (ε-fraction are A and B tokens or no minority)
  ↔ cancellation phase

- Extinction time
  (no minority tokens left)



Key ingredient: annihilation dynamics

- Annihilation dynamics:
  - A and B tokens meet → annihilation
  - Incident edge of token sampled → token moves

- ε-clearing time

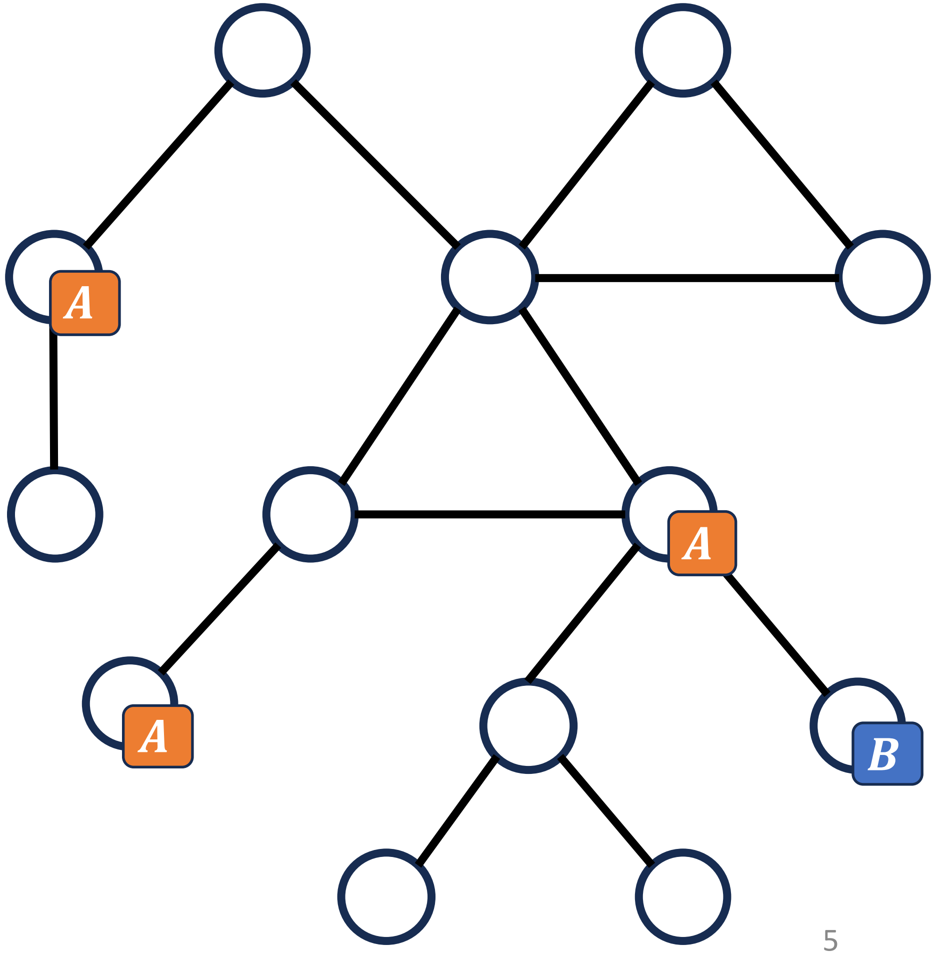
  (ε-fraction are A and B tokens or no minority)

  ↔ cancellation phase

- Extinction time

  (no minority tokens left)

  ↔ doubling phase



Key ingredient: annihilation dynamics

- Annihilation dynamics:
  - A and B tokens meet → annihilation
  - Incident edge of token sampled → token moves

- ε-clearing time

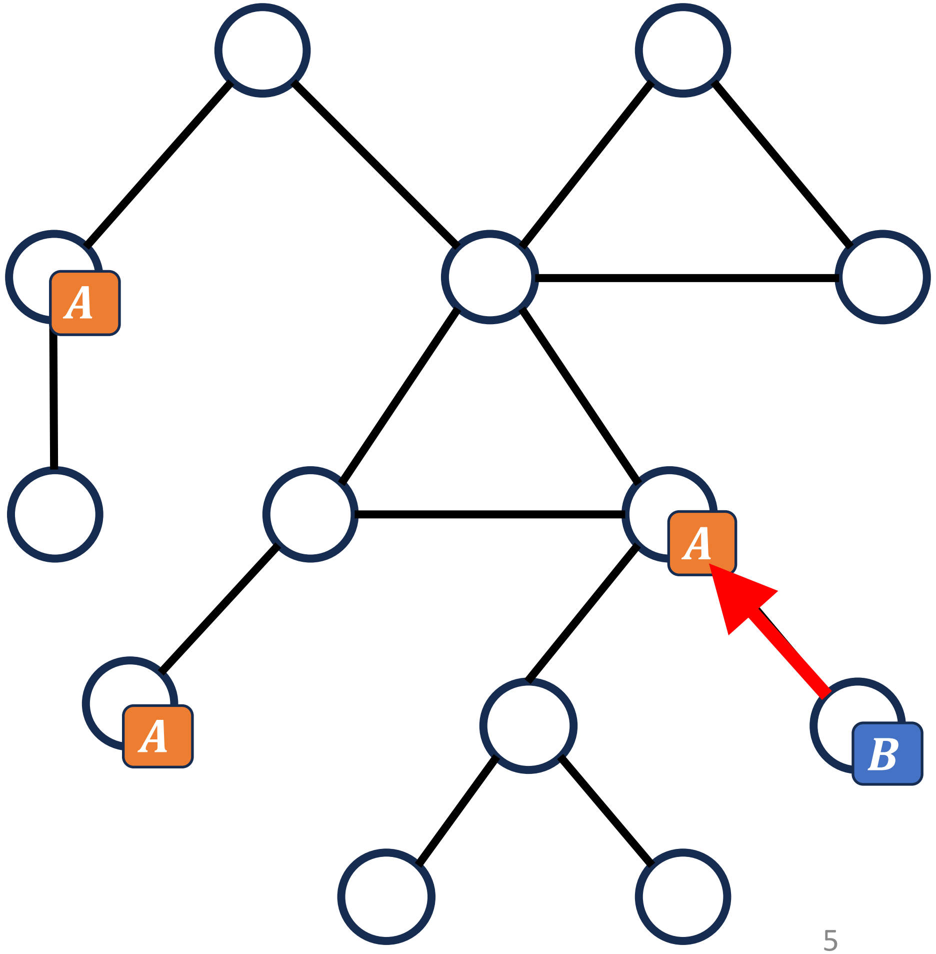
  (ε-fraction are A and B tokens or no minority)

  ↔ cancellation phase

- Extinction time

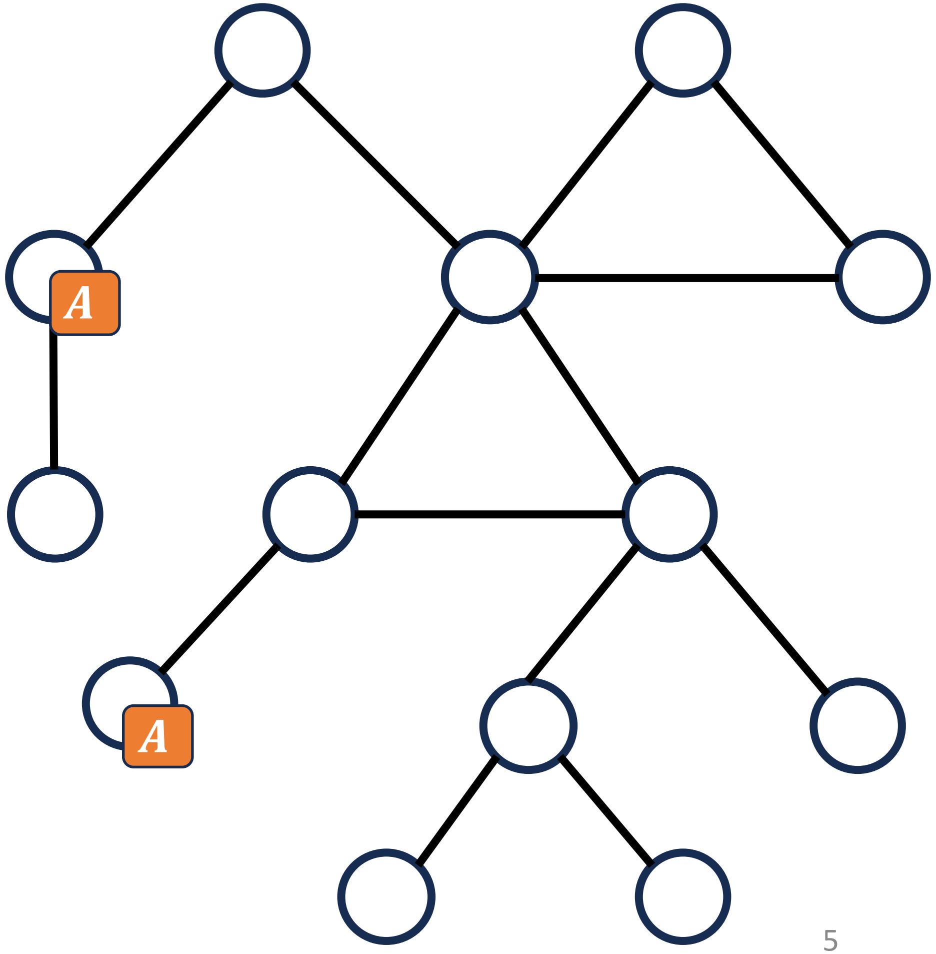
  (no minority tokens left)

  ↔ doubling phase

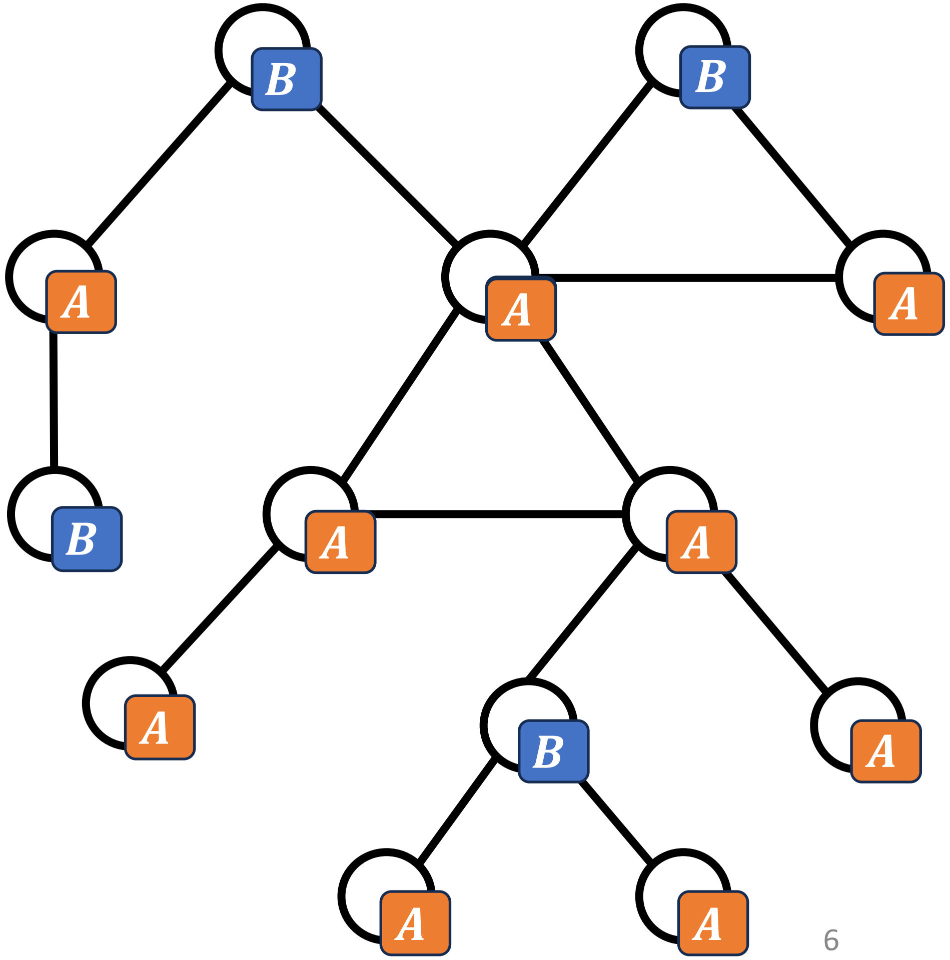


Key ingredient: annihilation dynamics

- Annihilation dynamics:
  - A and B tokens meet → annihilation
  - Incident edge of token sampled → token moves
- ε-clearing time
  (ε-fraction are A and B tokens or no minority)
  ↔ cancellation phase
- Extinction time
  (no minority tokens left)
  ↔ doubling phase

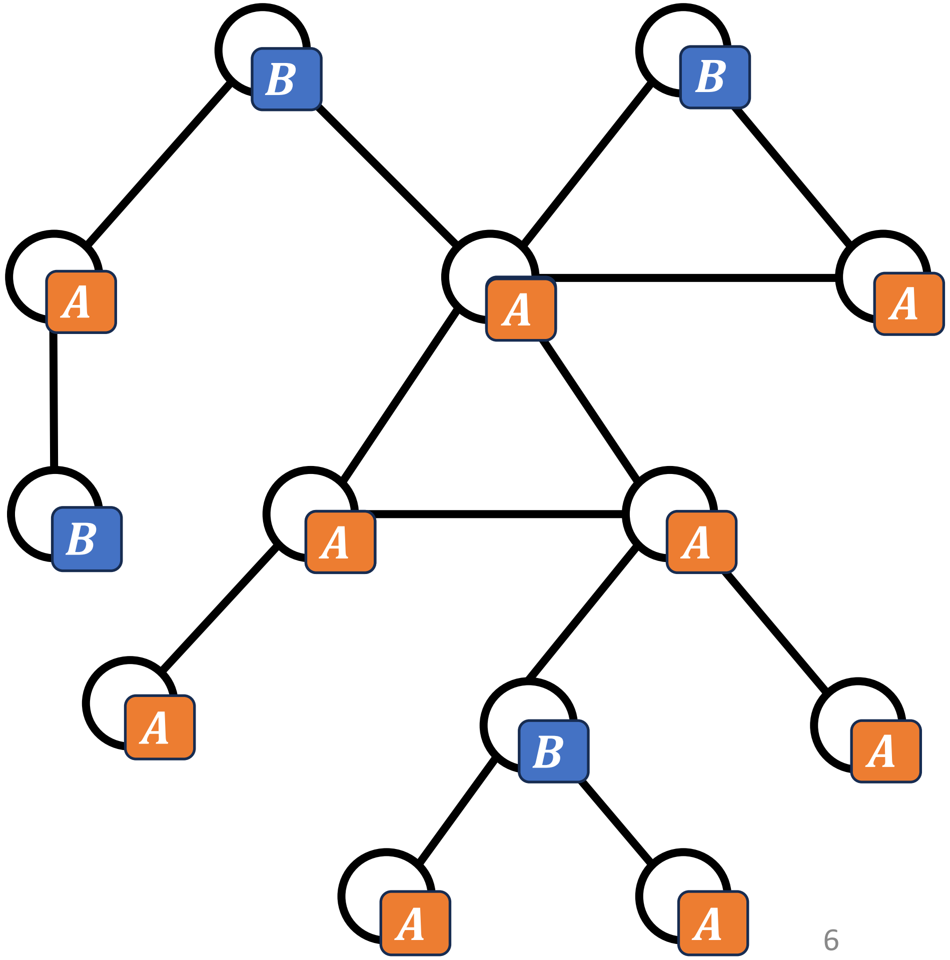


Results for annihilation dynamics



Results for annihilation dynamics

- Parameters:

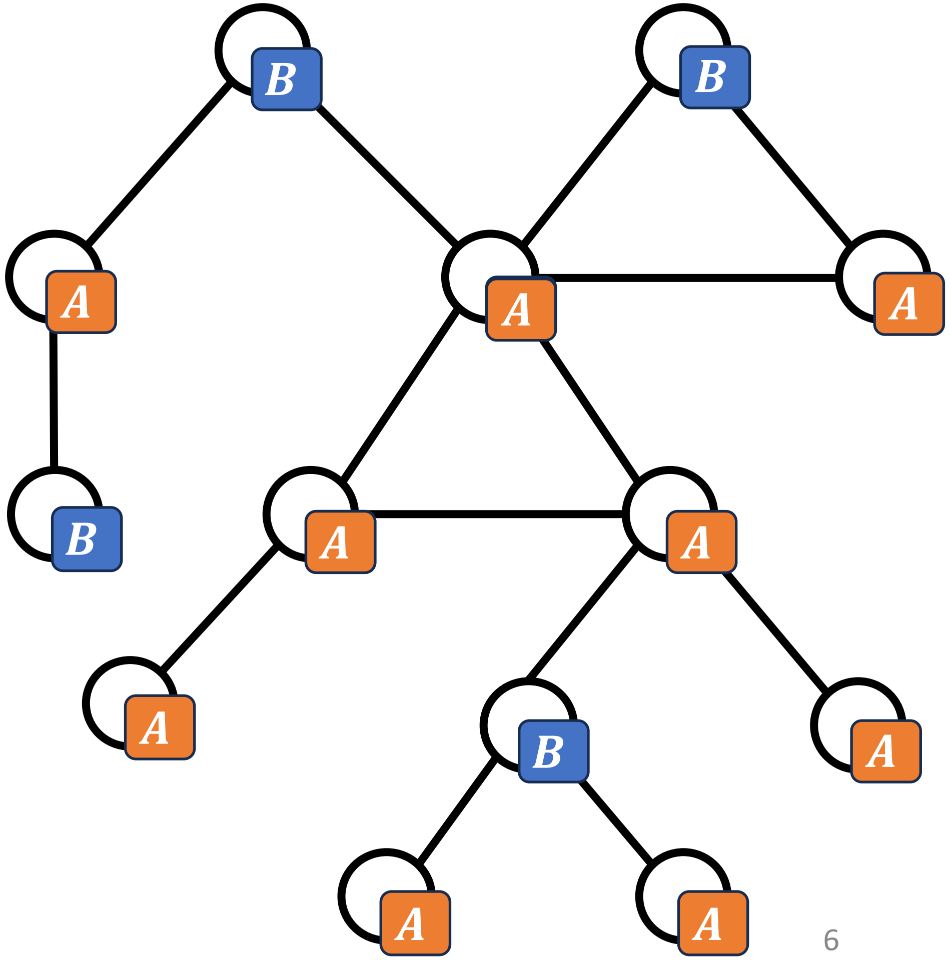


$$\gamma = \frac{|8-5|}{13} = \frac{3}{13}$$

Results for annihilation dynamics

- Parameters:
  - normalized bias $\gamma = |A_0 - B_0|/n$

    (normalized difference of initial tokens)

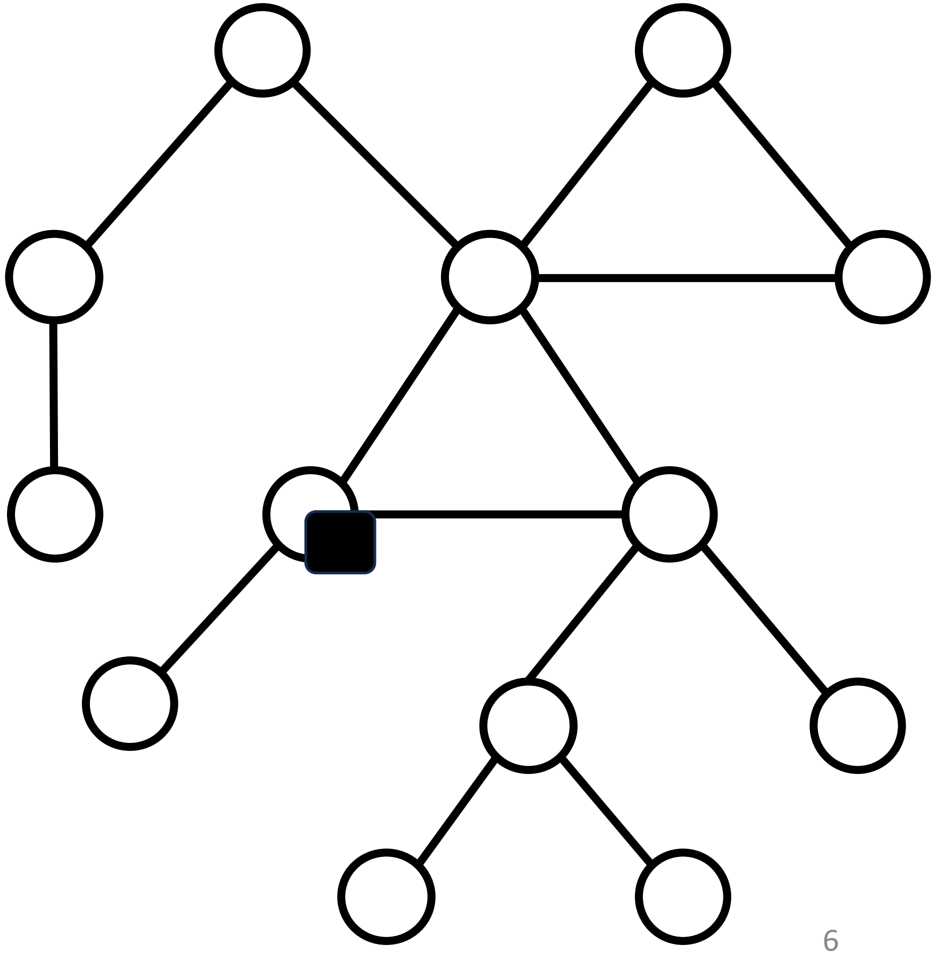


Results for annihilation dynamics

- Parameters:
  - normalized bias $\gamma = |A_0 - B_0|/n$

    (normalized difference of initial tokens)
  - relaxation time τ_{rel}

    (lazy token random walk)

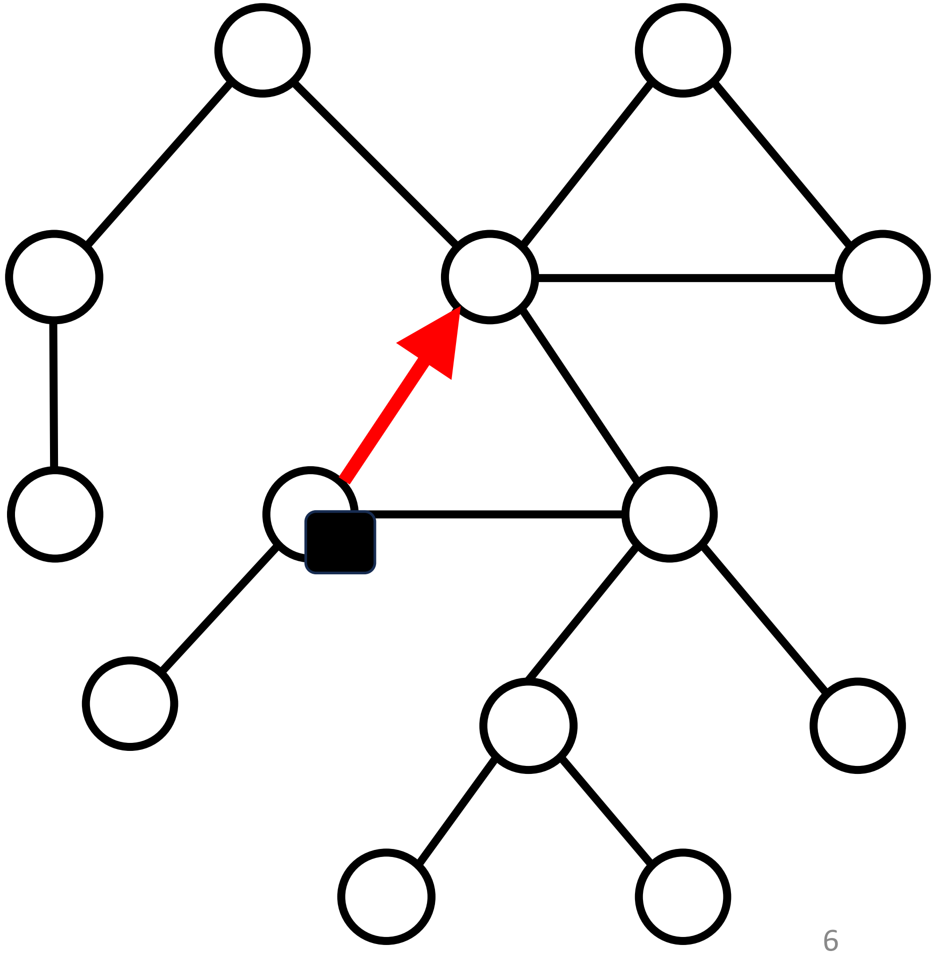


Results for annihilation dynamics

- Parameters:
  - normalized bias $\gamma = |A_0 - B_0|/n$

    (normalized difference of initial tokens)
  - relaxation time τ_{rel}

    (lazy token random walk)

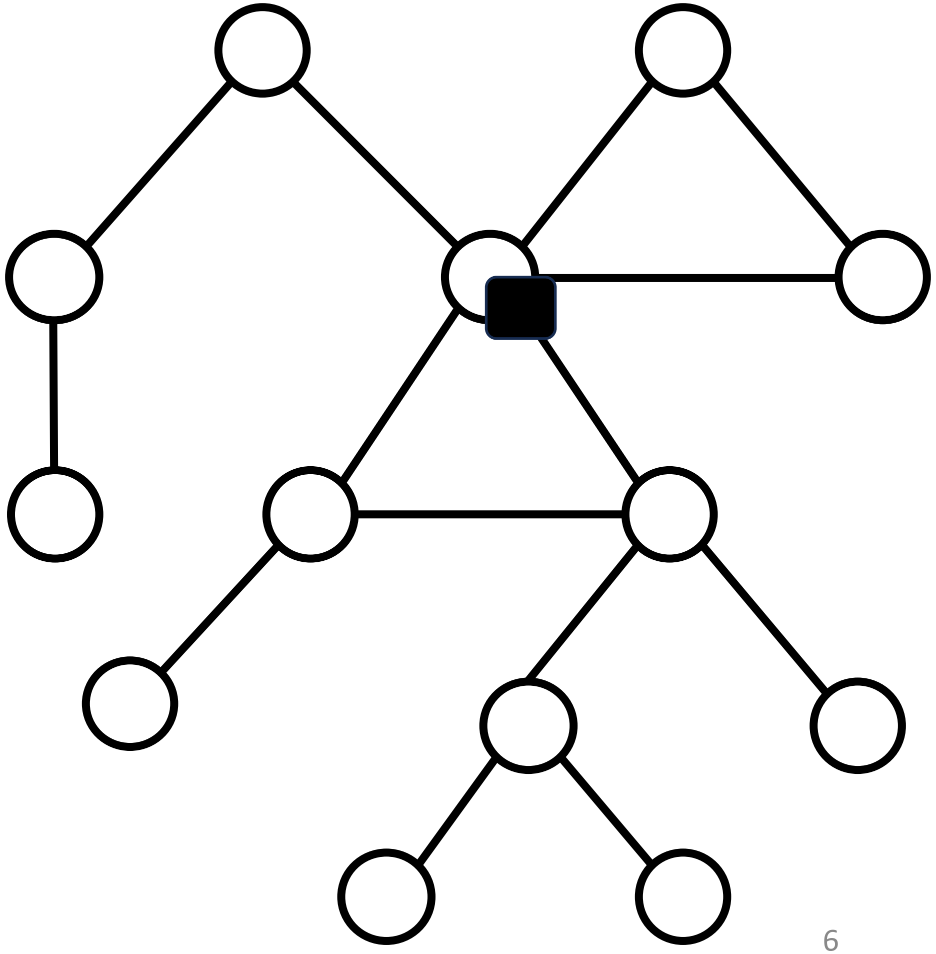


Results for annihilation dynamics

- Parameters:
  - normalized bias $\gamma = |A_0 - B_0|/n$

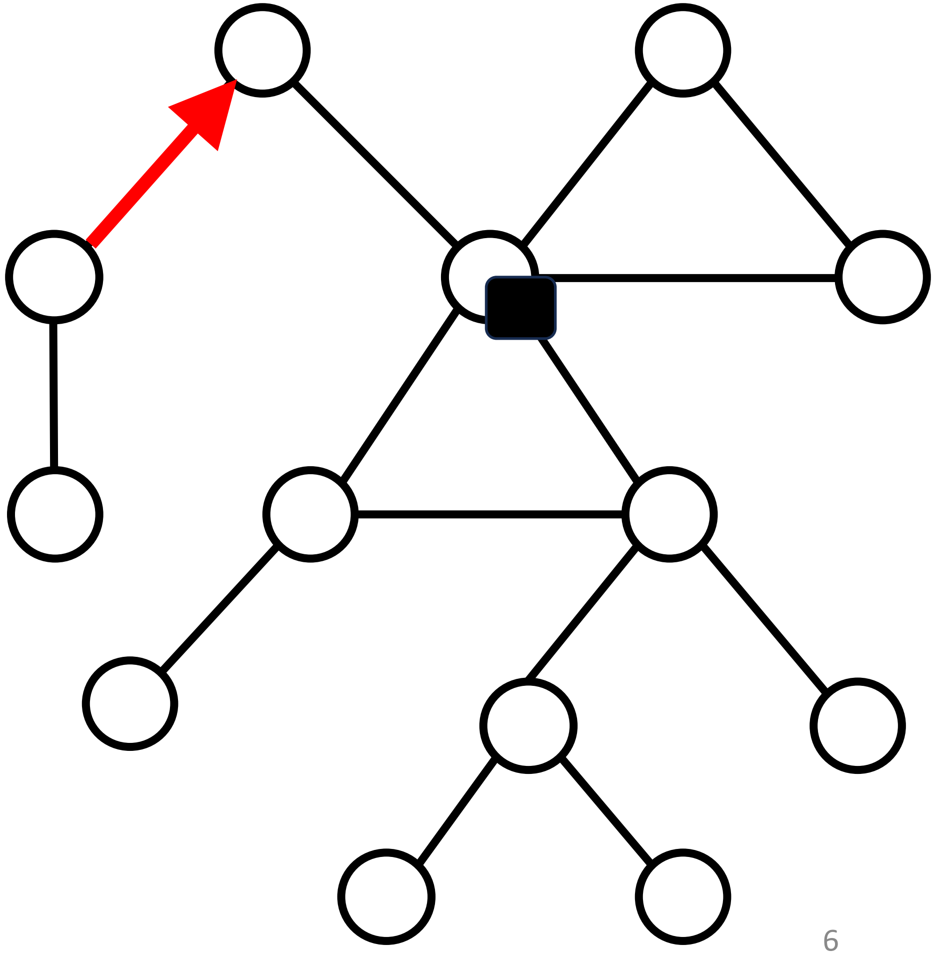
    (normalized difference of initial tokens)
  - relaxation time τ_{rel}

    (lazy token random walk)



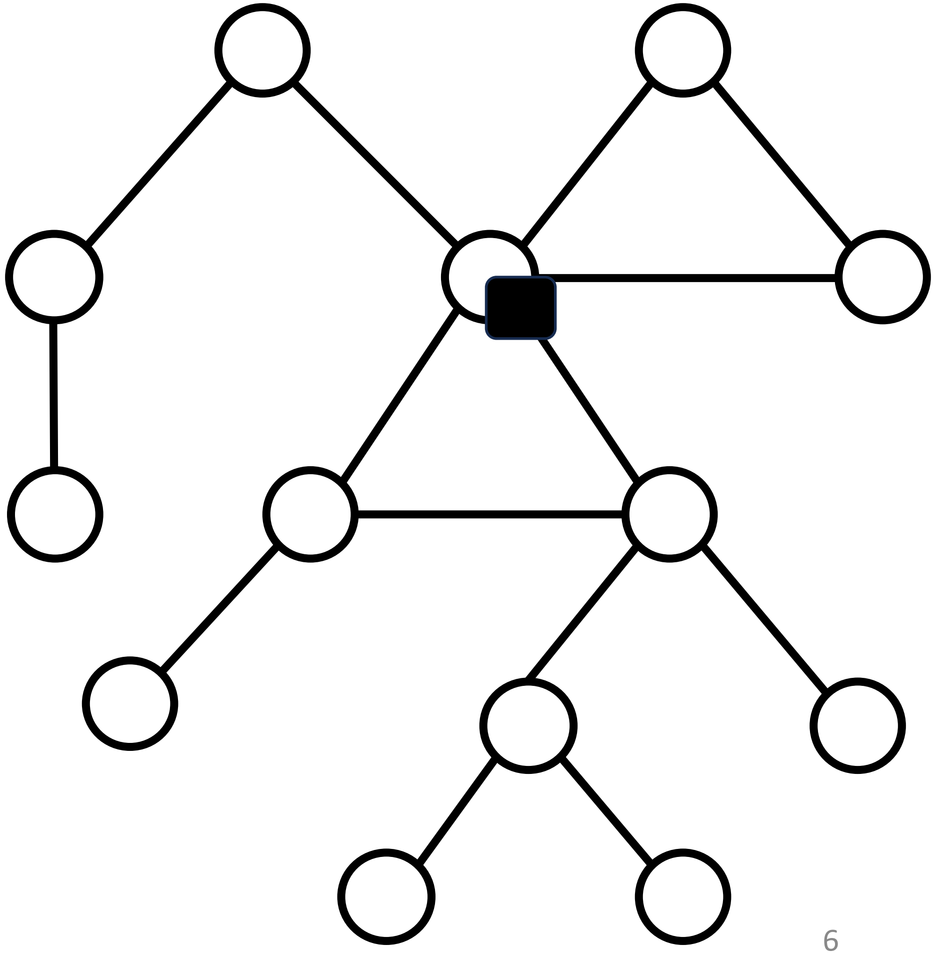
Results for annihilation dynamics

- Parameters:
  - normalized bias $\gamma = |A_0 - B_0|/n$
    (normalized difference of initial tokens)
  - relaxation time τ_{rel}
    (lazy token random walk)



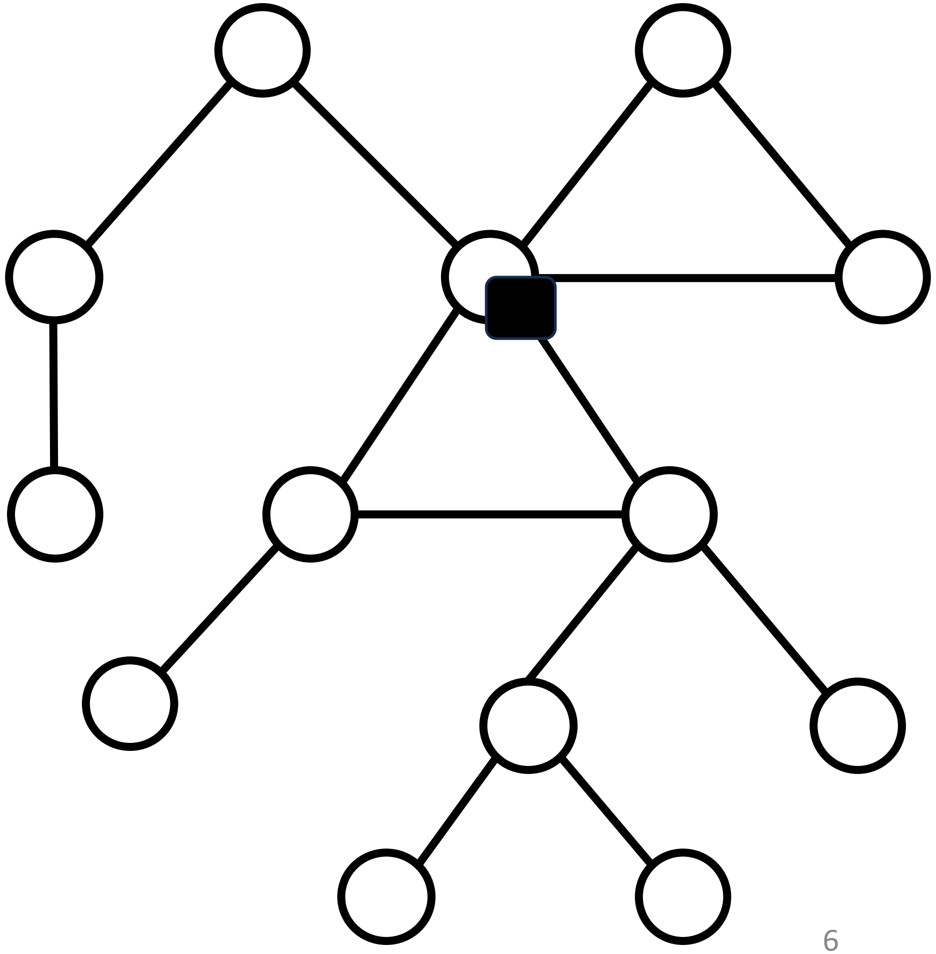
Results for annihilation dynamics

- Parameters:
  - normalized bias $\gamma = |A_0 - B_0|/n$
    (normalized difference of initial tokens)
  - relaxation time τ_{rel}
    (lazy token random walk)



Results for annihilation dynamics

- Parameters:
  - normalized bias $\gamma = |A_0 - B_0|/n$
    (normalized difference of initial tokens)
  - relaxation time τ_{rel}
    (lazy token random walk)
- We obtain that w.h.p. and in expectation:
  - Extinction time in $O(\tau_{rel}/\gamma \log n)$
    (no minority tokens left)



Results for annihilation dynamics

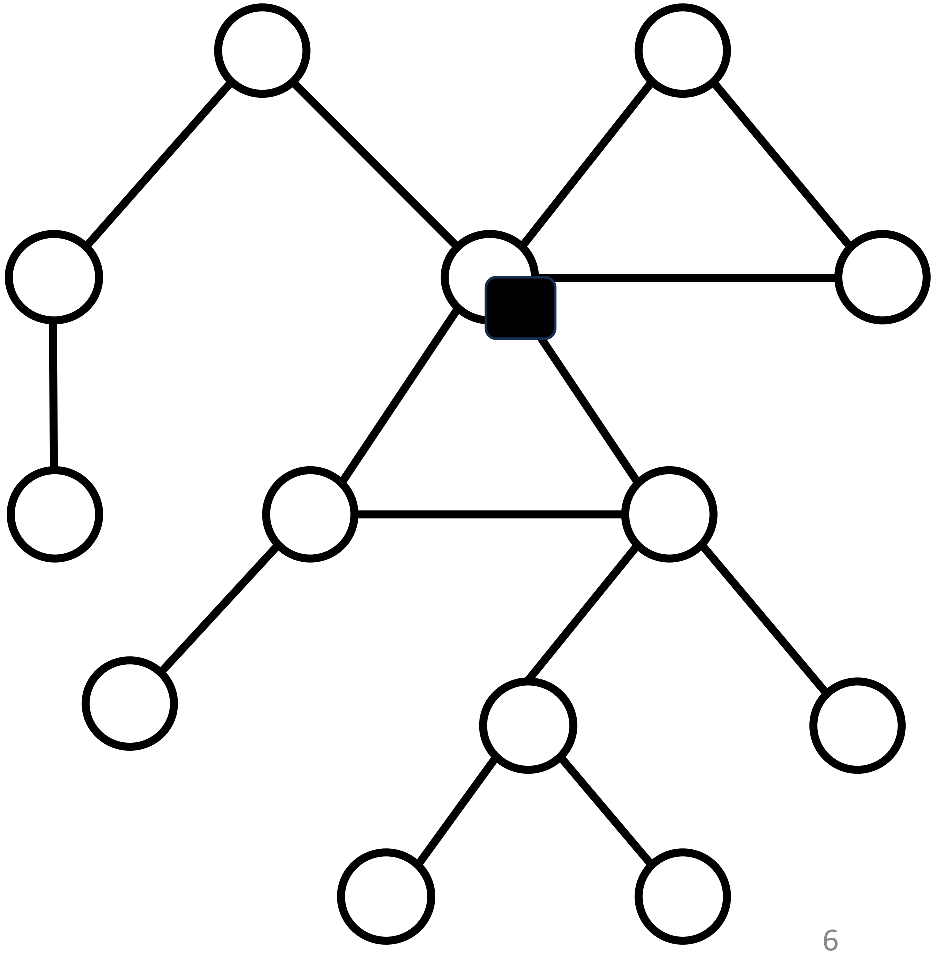
- Parameters:
  - normalized bias $\gamma = |A_0 - B_0|/n$

    (normalized difference of initial tokens)
  - relaxation time τ_{rel}

    (lazy token random walk)
- We obtain that w.h.p. and in expectation:
  - Extinction time in $O(\tau_{rel}/\gamma \log n)$

    (no minority tokens left)
  - ε-clearing time in $O(\tau_{rel}/\varepsilon \log n)$

    (ε-fraction of A and B tokens left)



New results: 4-state protocol

New results: 4-state protocol

Theorem
The 4-state protocol stabilizes in $O(\tau_{rel}/\gamma \log n)$ steps w.h.p. and in expectation.

New results: 4-state protocol

Theorem
The 4-state protocol stabilizes in $O(\tau_{rel}/\gamma \log n)$ steps w.h.p. and in expectation.

- Comparison to state of the art:

New results: 4-state protocol

Theorem
The 4-state protocol stabilizes in $O(\tau_{rel}/\gamma \log n)$ steps w.h.p. and in expectation.

- Comparison to state of the art:

Graph	Stabilization time	Number of states	Reference
Clique	$O(n/\gamma \log n)$	4	Draief and Vojnović, INFOCOM'10

New results: 4-state protocol

Theorem

The 4-state protocol stabilizes in $O(\tau_{rel}/\gamma \log n)$ steps w.h.p. and in expectation.

- Comparison to state of the art:

Graph	Stabilization time	Number of states	Reference
Clique	$O(n/\gamma \log n)$	4	Draief and Vojnović, INFOCOM'10
Regular Expanders	$O(n/\gamma \log n)$	4	**This work**

New results: 4-state protocol

Theorem
The 4-state protocol stabilizes in $O(\tau_{rel}/\gamma \log n)$ steps w.h.p. and in expectation.

- Comparison to state of the art:

Graph	Stabilization time	Number of states	Reference
Clique	$O(n/\gamma \log n)$	4	Draief and Vojnović, INFOCOM'10
Regular Expanders	$O(n/\gamma \log n)$	4	**This work**
Connected	$O(n^6)$	4	Mertzios et al., ICALP'14
Connected	$O(\log n \ /\lambda(\gamma))$	4	Draief and Vojnović, INFOCOM'10
Connected	$O(\tau_{rel}/\gamma \log n)$	4	**This work**

New results: fast protocol

Δ = maximum degree
δ = minimum degree

New results: fast protocol

Theorem (general version)
There exists a protocol that stabilizes in $O(\Delta/\delta\tau_{rel} \log n \log 1/\gamma)$ steps w.h.p. and in expectation using $O(\log n\,(\log(\Delta/\delta) + \log(\tau_{rel}/n)))$ states.

Δ = maximum degree
δ = minimum degree

New results: fast protocol

Theorem (general version)
There exists a protocol that stabilizes in $O(\Delta/\delta\tau_{rel} \log n \log 1/\gamma)$ steps w.h.p. and in expectation using $O(\log n\, (\log(\Delta/\delta) + \log(\tau_{rel}/n)))$ states.

Graph	Stabilization time	Number of states	Reference
Clique	$\Theta(n \log n)$	$\Theta(\log n)$	Doty et al., FOCS’22
Regular expanders	$O(n\log^2 n)$	$\Theta(\log n)$	**This work**

Δ = maximum degree
δ = minimum degree

New results: fast protocol

Theorem (general version)
There exists a protocol that stabilizes in $O(\Delta/\delta\tau_{rel}\log n\log 1/\gamma)$ steps w.h.p. and in expectation using $O(\log n\,(\log(\Delta/\delta)+\log(\tau_{rel}/n)))$ states.

Graph	Stabilization time	Number of states	Reference
Clique	$\Theta(n\log n)$	$\Theta(\log n)$	Doty et al., FOCS'22
Regular expanders	$O(n\log^2 n)$	$\Theta(\log n)$	**This work**
Regular	$O(\phi^{-2}n\log^6 n)$	$O(\phi^{-2}\log^5 n)$	Alistarh et al., OPODIS'21
Regular	$O(\phi^{-2}n\log^2 n)$	$O(\log\phi^{-1}\log n)$	**This work**

New results: lower bounds

New results: lower bounds

Theorem

There is no protocol faster than $\Omega(\max\{D, \log n\} \cdot n)$ for solving exact majority on a regular graph G, with n nodes and diameter D.

New results: lower bounds

Theorem
There is no protocol faster than $\Omega(\max\{D, \log n\} \cdot n)$ for solving exact majority on a regular graph G, with n nodes and diameter D.

Theorem
There is no protocol faster than $\Omega(D \cdot m)$ for solving exact majority on an arbitrary graph G, with m edges and diameter D.

Summary

Graph	Stabilization time	Number of states	Reference
Regular Expanders	$O(n/\gamma \log n)$	4	**This work**
Clique	$O(n^{2-\varepsilon})$	$\Omega(\log n)$	Alistarh et al., SODA'18
Clique	$\Theta(n \log n)$	$\Theta(\log n)$	Doty et al., FOCS'22

Summary

Graph	Stabilization time	Number of states	Reference
Regular Expanders	$O(n/\gamma \log n)$	4	**This work**
Clique	$O(n^{2-\varepsilon})$	$\Omega(\log n)$	Alistarh et al., SODA'18
Clique	$\Theta(n \log n)$	$\Theta(\log n)$	Doty et al., FOCS'22
Regular	$O(\tau_{rel} \log n \log 1/\gamma)$	$O(\log n \log \tau_{rel}/n)$	**This work**

Summary

Graph	Stabilization time	Number of states	Reference
Regular Expanders	$O(n/\gamma \log n)$	4	**This work**
Clique	$O(n^{2-\varepsilon})$	$\Omega(\log n)$	Alistarh et al., SODA'18
Clique	$\Theta(n \log n)$	$\Theta(\log n)$	Doty et al., FOCS'22
Regular	$O(\tau_{rel} \log n \log 1/\gamma)$	$O(\log n \log \tau_{rel}/n)$	**This work**
Regular	$\Omega(\max\{D, \log n\} \cdot n)$	any	**This work**

Summary

Graph	Stabilization time	Number of states	Reference
Regular Expanders	$O(n/\gamma \log n)$	4	**This work**
Clique	$O(n^{2-\varepsilon})$	$\Omega(\log n)$	Alistarh et al., SODA'18
Clique	$\Theta(n \log n)$	$\Theta(\log n)$	Doty et al., FOCS'22
Regular	$O(\tau_{rel} \log n \log 1/\gamma)$	$O(\log n \log \tau_{rel}/n)$	**This work**
Regular	$\Omega(\max\{D, \log n\} \cdot n)$	any	**This work**
Connected	$O(n^6)$	4	Mertzios et al., ICALP'14
Connected	$O(\log n \ /\lambda(\gamma))$	4	Draief and Vojnović, INFOCOM'10
Connected	$O(\tau_{rel}/\gamma \log n)$	4	**This work**

Summary

Graph	Stabilization time	Number of states	Reference
Regular Expanders	$O(n/\gamma \log n)$	4	**This work**
Clique	$O(n^{2-\varepsilon})$	$\Omega(\log n)$	Alistarh et al., SODA'18
Clique	$\Theta(n \log n)$	$\Theta(\log n)$	Doty et al., FOCS'22
Regular	$O(\tau_{rel} \log n \log 1/\gamma)$	$O(\log n \log \tau_{rel}/n)$	**This work**
Regular	$\Omega(\max\{D, \log n\} \cdot n)$	any	**This work**
Connected	$O(n^6)$	4	Mertzios et al., ICALP'14
Connected	$O(\log n \ /\lambda(\gamma))$	4	Draief and Vojnović, INFOCOM'10
Connected	$O(\tau_{rel}/\gamma \log n)$	4	**This work**
Connected	$O(\Delta/\delta\tau_{rel} \log n \log 1/\gamma)$	$O(\log n\ (\log(\Delta/\delta) + \log(\tau_{rel}/n)))$	**This work**

Summary

Graph	Stabilization time	Number of states	Reference
Regular Expanders	$O(n/\gamma \log n)$	4	**This work**
Clique	$O(n^{2-\varepsilon})$	$\Omega(\log n)$	Alistarh et al., SODA'18
Clique	$\Theta(n \log n)$	$\Theta(\log n)$	Doty et al., FOCS'22
Regular	$O(\tau_{rel} \log n \log 1/\gamma)$	$O(\log n \log \tau_{rel}/n)$	**This work**
Regular	$\Omega(\max\{D, \log n\} \cdot n)$	any	**This work**
Connected	$O(n^6)$	4	Mertzios et al., ICALP'14
Connected	$O(\log n \ /\lambda(\gamma))$	4	Draief and Vojnović, INFOCOM'10
Connected	$O(\tau_{rel}/\gamma \log n)$	4	**This work**
Connected	$O(\Delta/\delta \tau_{rel} \log n \log 1/\gamma)$	$O(\log n \, (\log(\Delta/\delta) + \log(\tau_{rel}/n)))$	**This work**
Connected	$\Omega(D \cdot m)$	any	**This work**