Optimal Computation in Anonymous Dynamic Networks

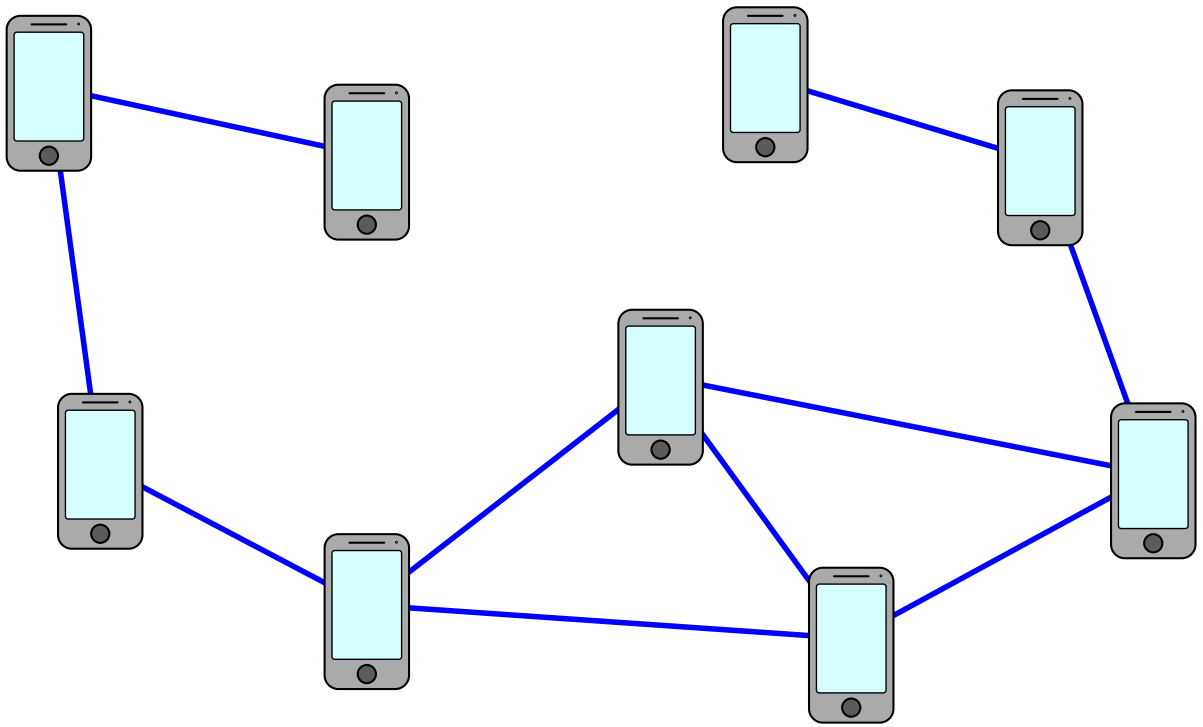
Giovanni Viglietta

Joint work with Giuseppe A. Di Luna

Pisa – September 21, 2022

Dynamic networks

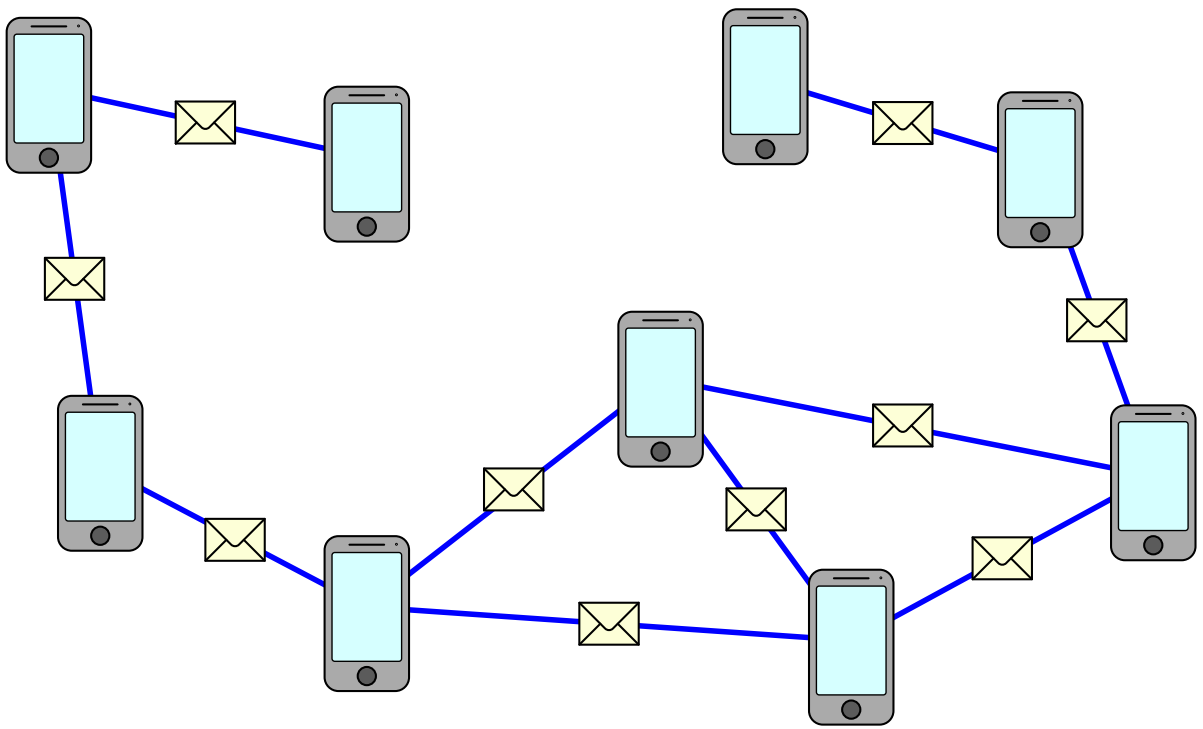
In a *dynamic network*, some machines (or agents) are connected with each other through links that may change over time.



What can be computed by this network, and in how many rounds?

Dynamic networks

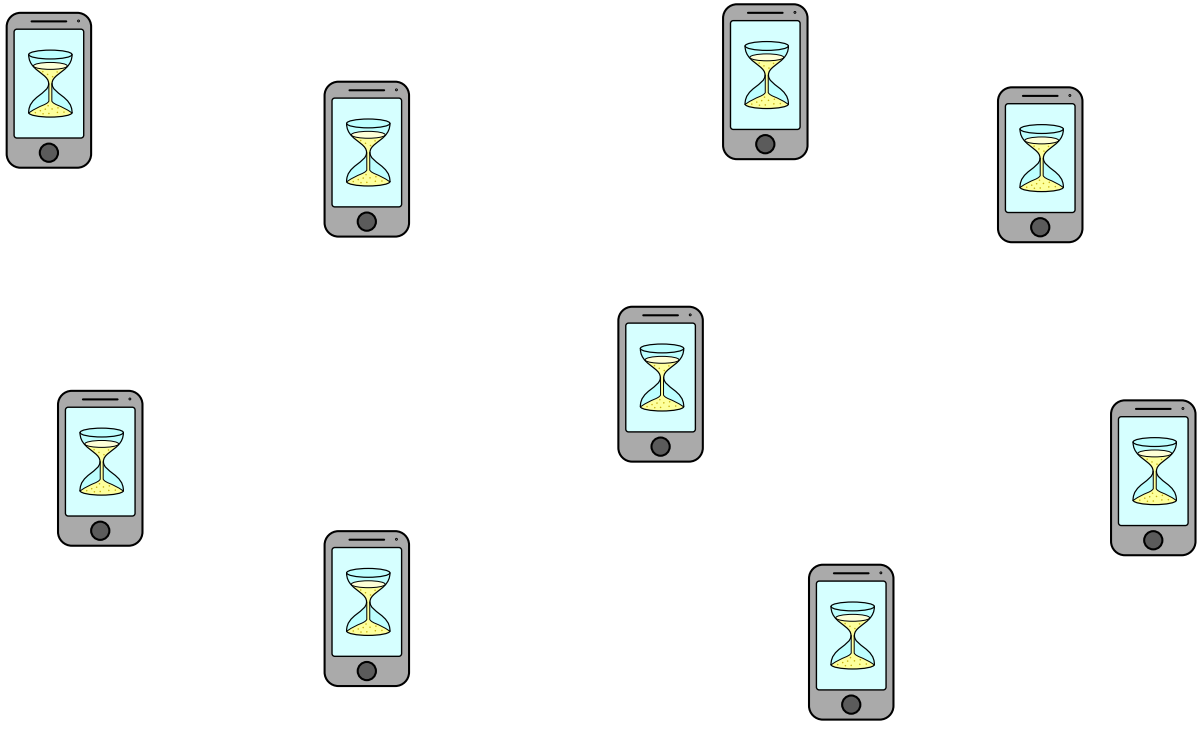
In a *dynamic network*, some machines (or agents) are connected with each other through links that may change over time.



What can be computed by this network, and in how many rounds?

Dynamic networks

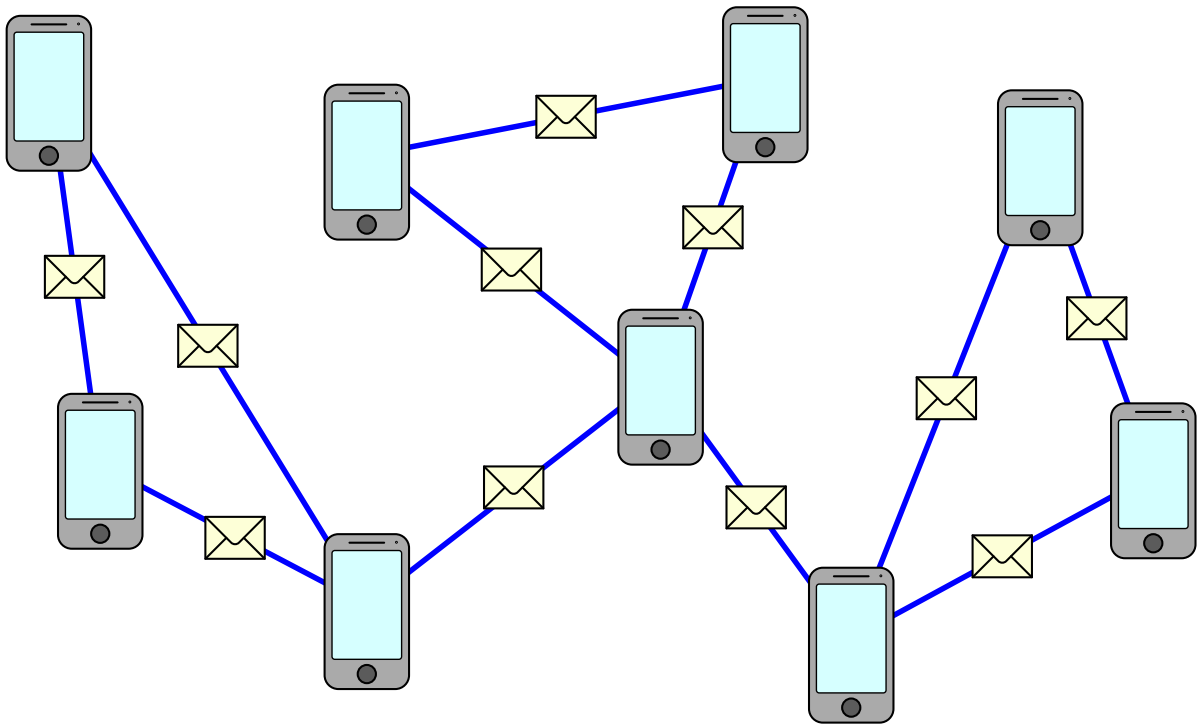
In a *dynamic network*, some machines (or agents) are connected with each other through links that may change over time.



What can be computed by this network, and in how many rounds?

Dynamic networks

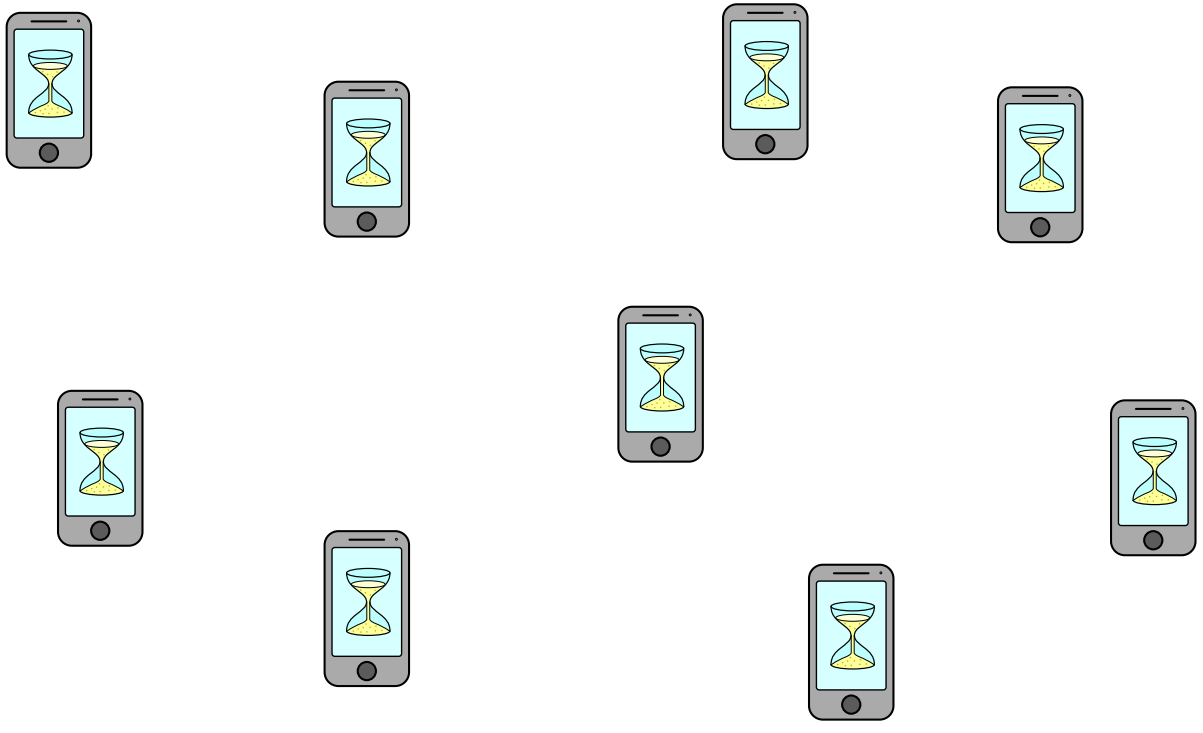
In a *dynamic network*, some machines (or agents) are connected with each other through links that may change over time.



What can be computed by this network, and in how many rounds?

Dynamic networks

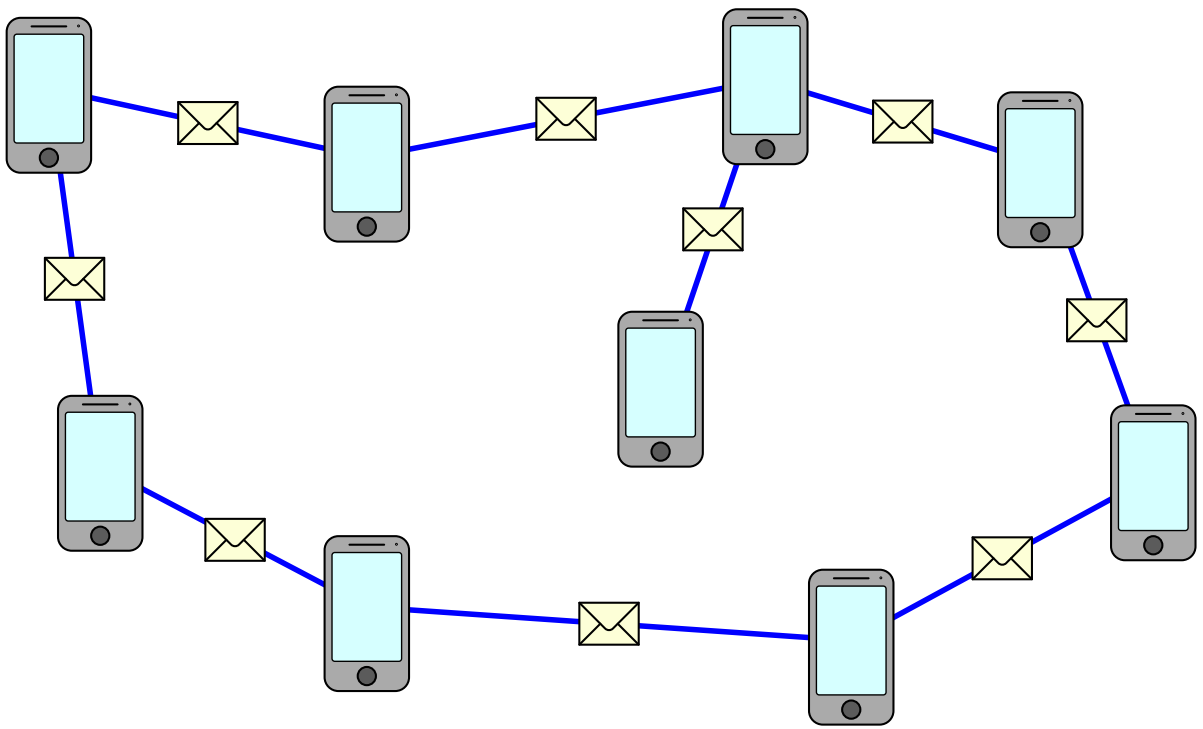
In a *dynamic network*, some machines (or agents) are connected with each other through links that may change over time.



What can be computed by this network, and in how many rounds?

Dynamic networks

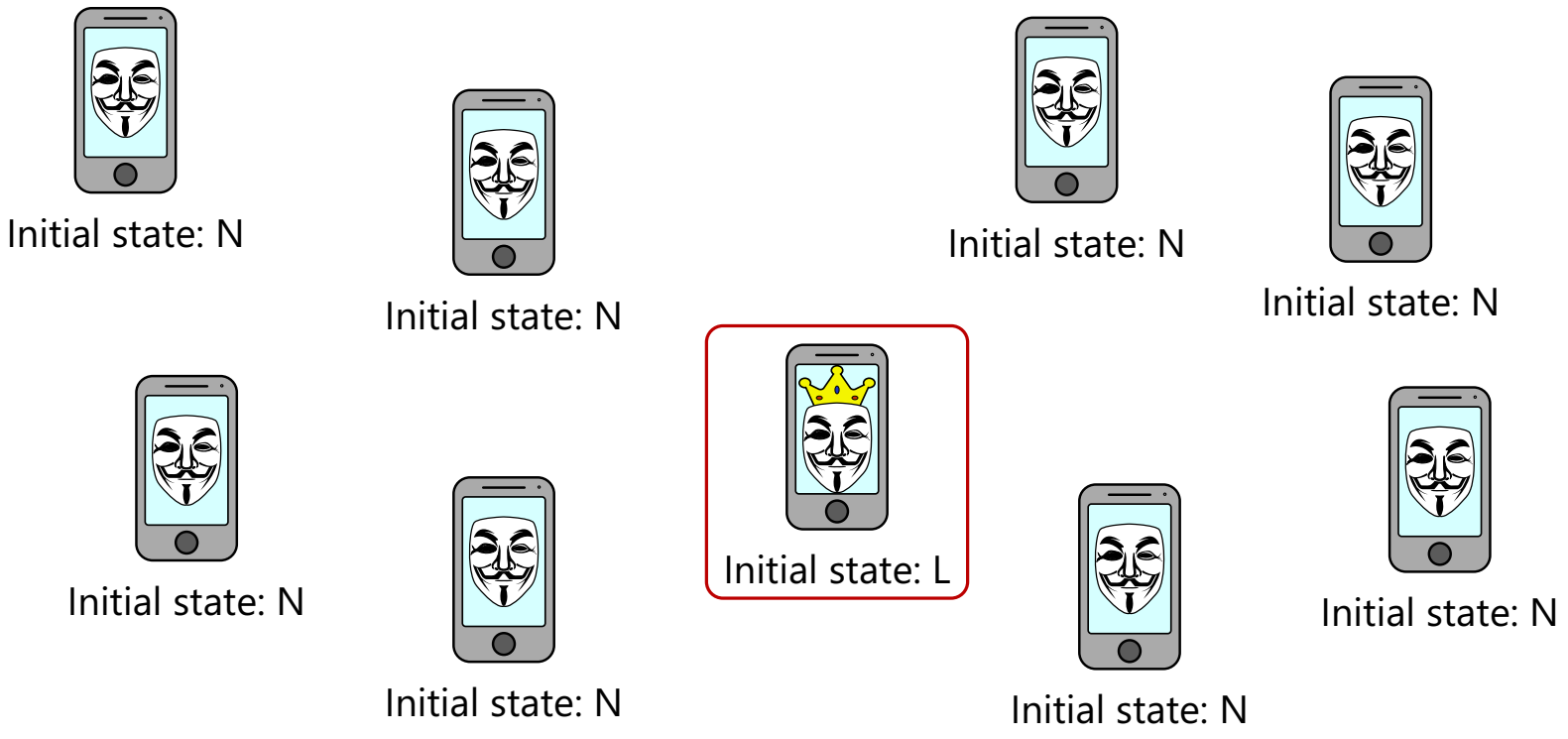
In a *dynamic network*, some machines (or agents) are connected with each other through links that may change over time.



What can be computed by this network, and in how many rounds?

Counting anonymous agents with a Leader

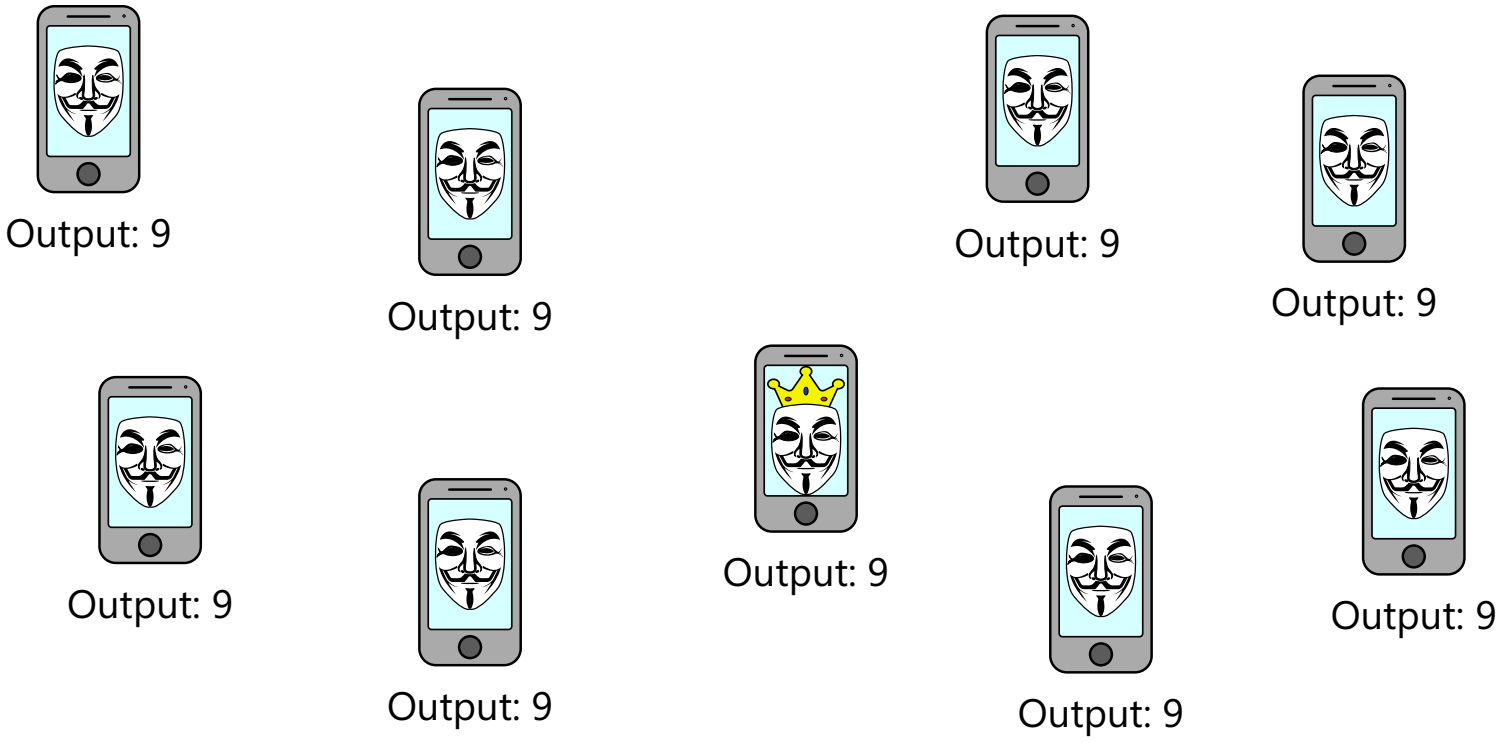
A common assumption is that the dynamic network is *anonymous*, i.e., all agents start in the same state, except one: the *Leader*.



The *complete problem* in this model is the **Counting Problem:** Eventually, all agents must know the total number of agents, n. (If agents have inputs, also compute how many agents have each input.)

Counting anonymous agents with a Leader

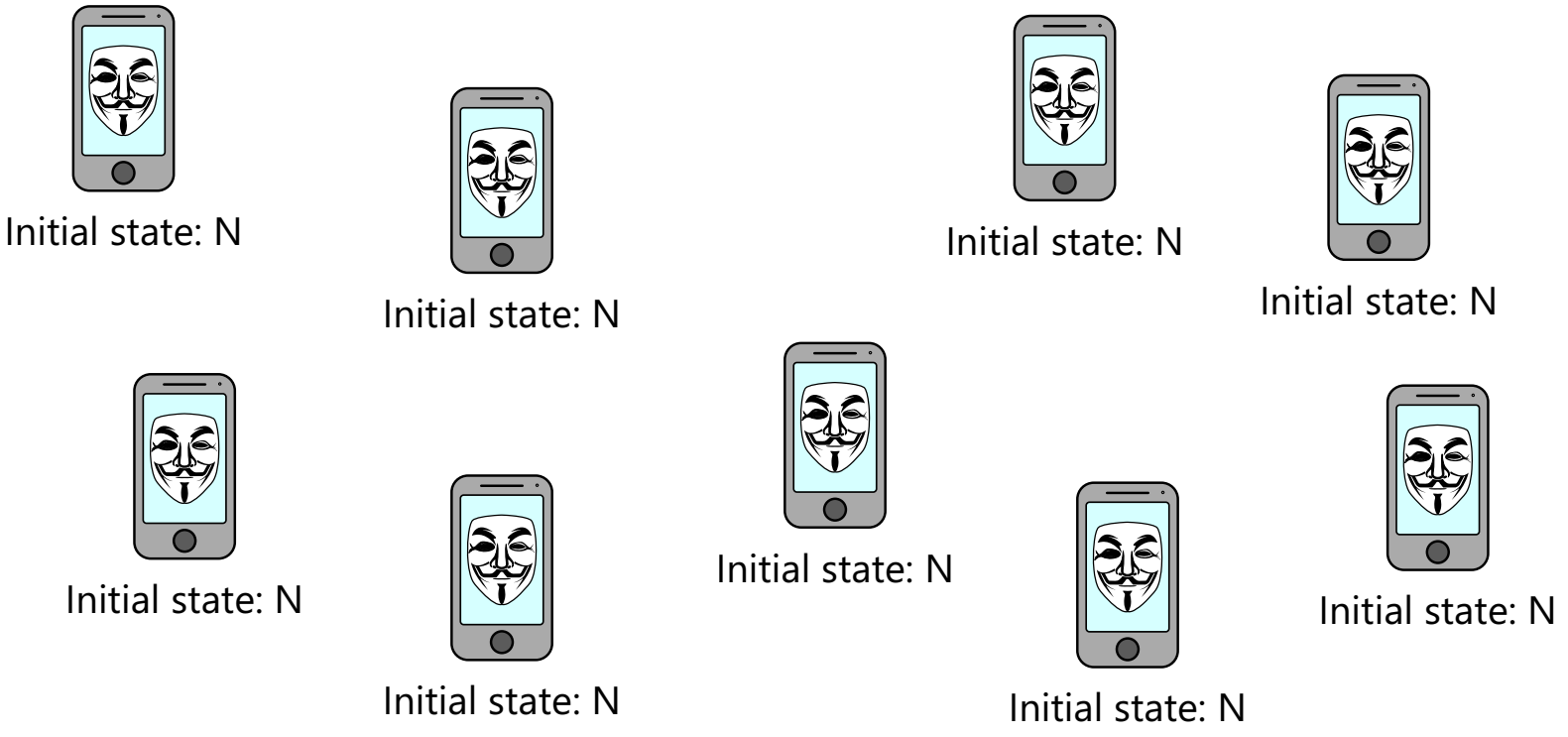
A common assumption is that the dynamic network is *anonymous*, i.e., all agents start in the same state, except one: the *Leader*.



The *complete problem* in this model is the **Counting Problem:** Eventually, all agents must know the total number of agents, n. (If agents have inputs, also compute how many agents have each input.)

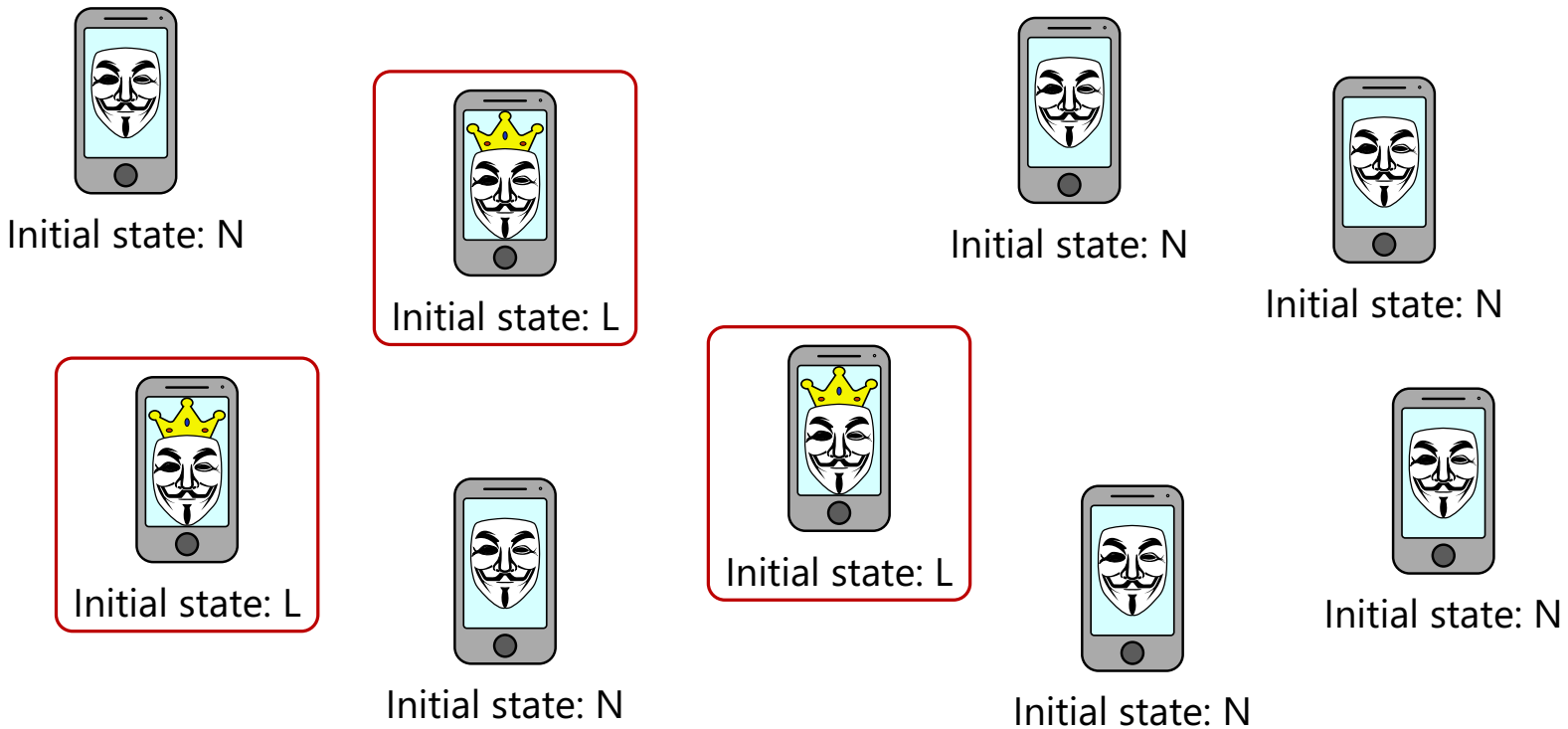
No Leader or multiple Leaders

Other interesting scenarios are those with *no Leader* or *multiple Leaders* (where the number of Leaders ℓ is known).



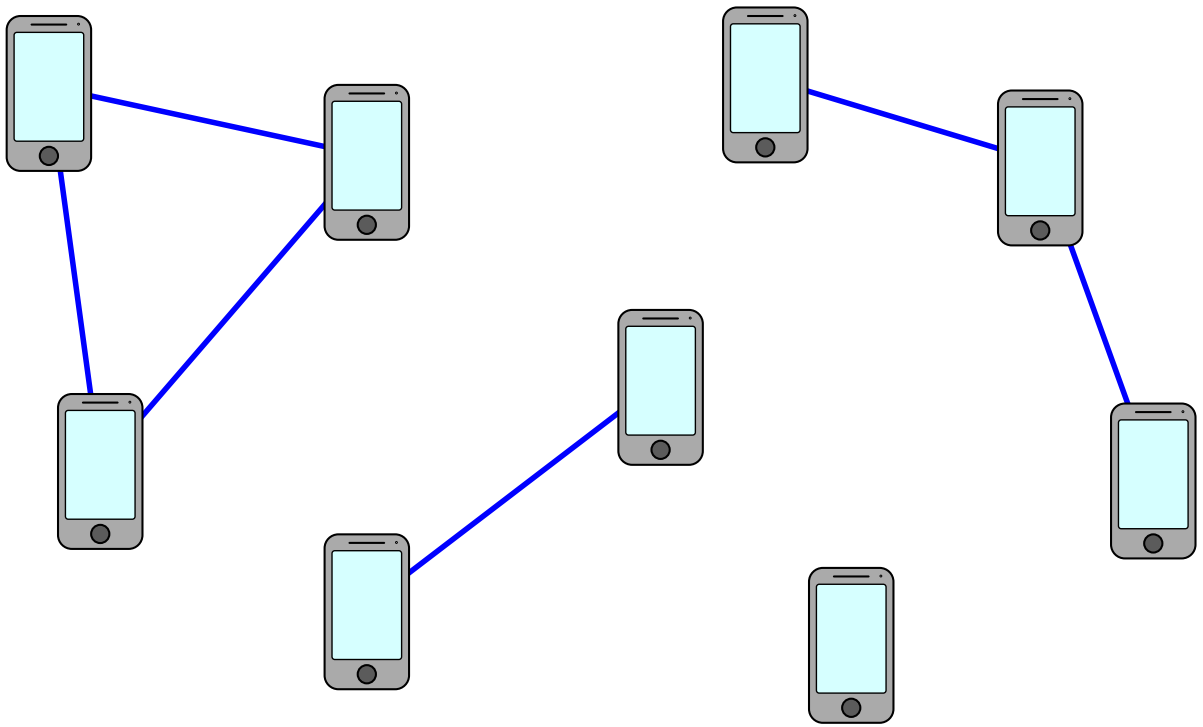
No Leader or multiple Leaders

Other interesting scenarios are those with *no Leader* or *multiple Leaders* (where the number of Leaders ℓ is known).



Disconnected networks

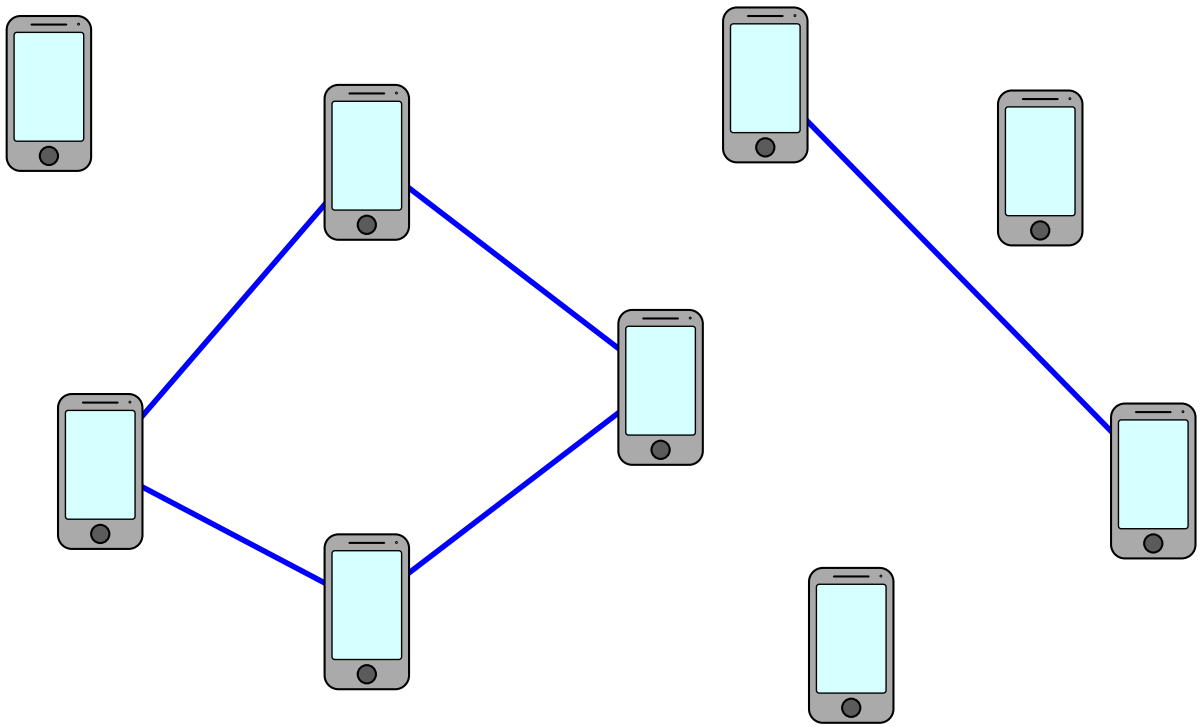
Also, while the network is typically assumed to be connected at every round, we may relax this assumption.



In a T-*interval-disconnected network*, the union of the network graphs at T consecutive rounds is a connected multi-graph.

Disconnected networks

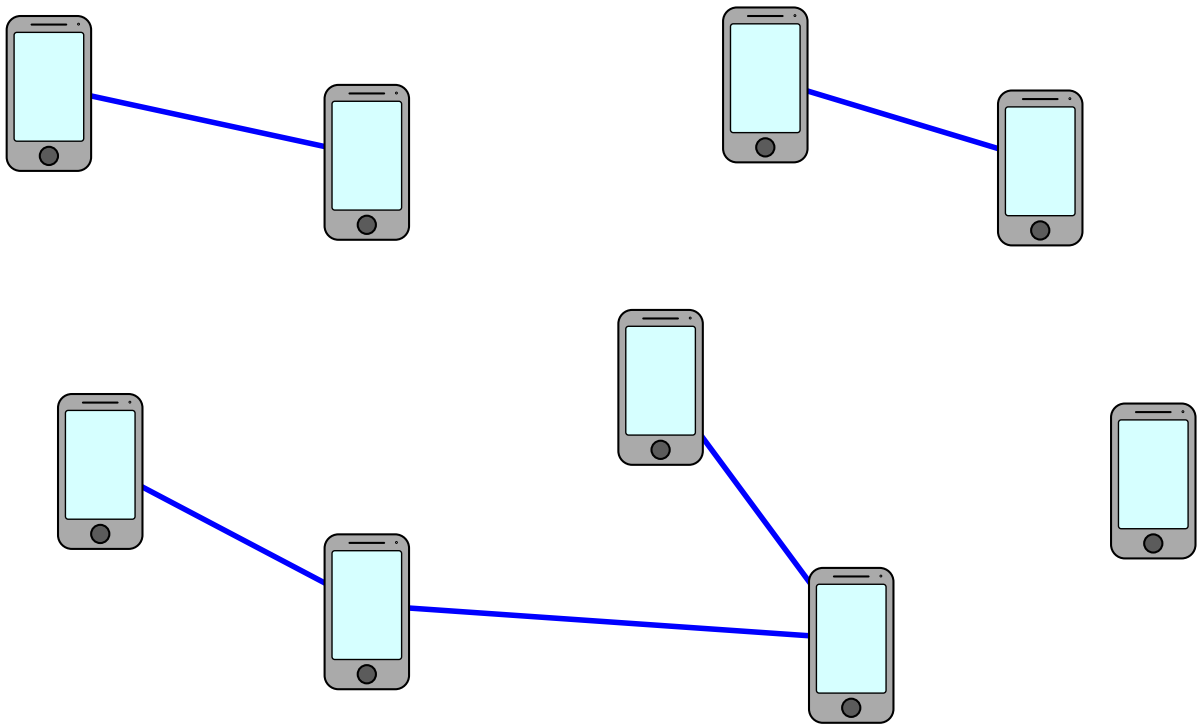
Also, while the network is typically assumed to be connected at every round, we may relax this assumption.



In a T-*interval-disconnected network*, the union of the network graphs at T consecutive rounds is a connected multi-graph.

Disconnected networks

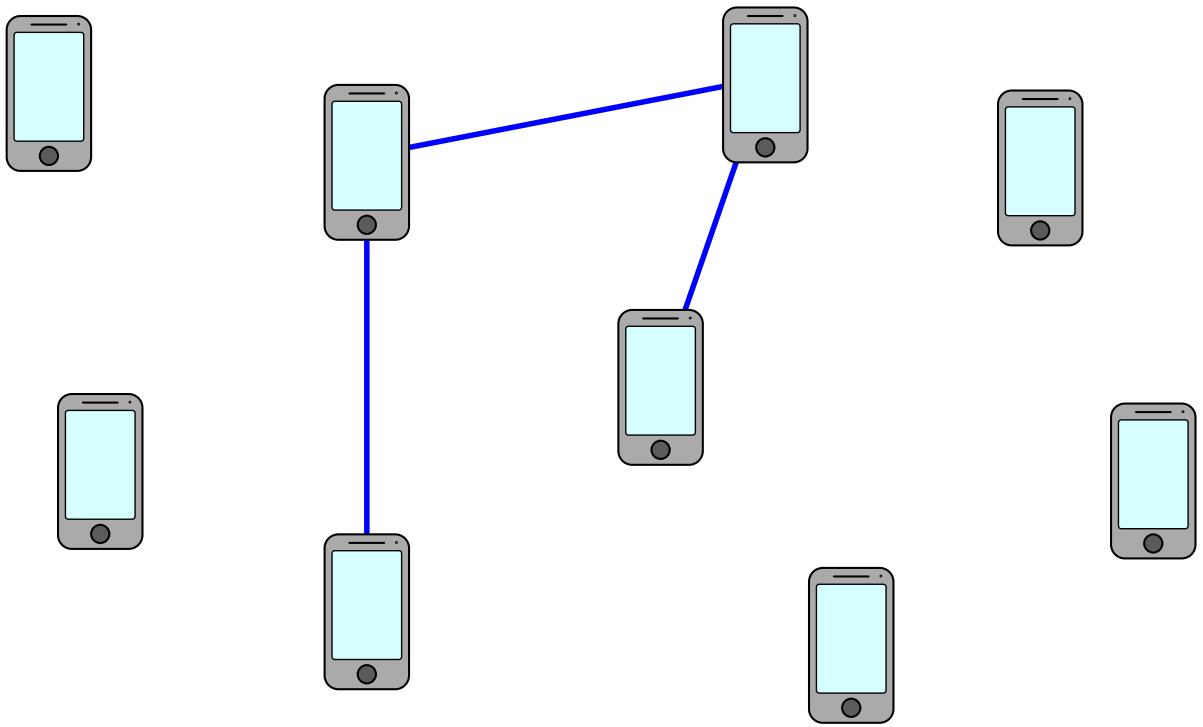
Also, while the network is typically assumed to be connected at every round, we may relax this assumption.



In a T-*interval-disconnected network*, the union of the network graphs at T consecutive rounds is a connected multi-graph.

Disconnected networks

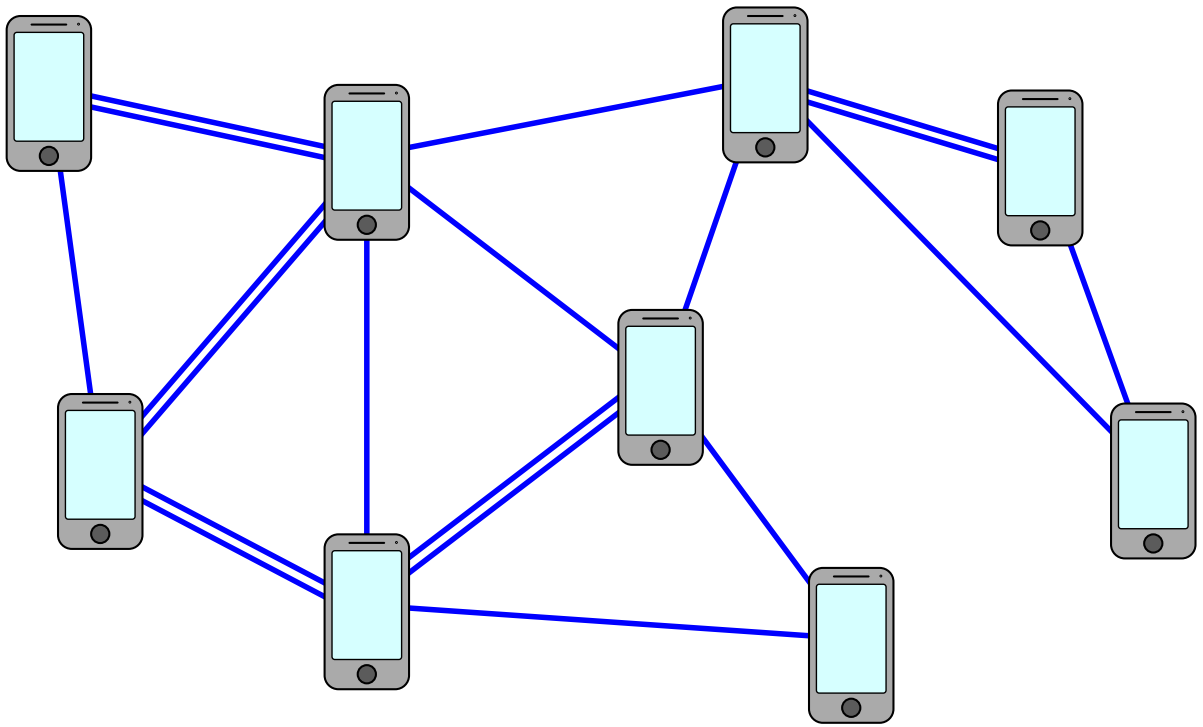
Also, while the network is typically assumed to be connected at every round, we may relax this assumption.



In a T-*interval-disconnected network*, the union of the network graphs at T consecutive rounds is a connected multi-graph.

Disconnected networks

Also, while the network is typically assumed to be connected at every round, we may relax this assumption.



In a T-*interval-disconnected network*, the union of the network graphs at T consecutive rounds is a connected multi-graph.

Literature on the Counting Problem

- **Michail et al.:** Looks impossible! (SSS 2013)
- **Di Luna et al.:** Solvable in $O(e^{N^2}N^3)$ rounds (ICDCN 2014)
- **Di Luna–Baldoni:** $O(n^{n+4})$ rounds (OPODIS 2015)
- **Kowalski–Mosteiro:** $O(n^5 \log^2 n)$ rounds (ICALP 2018 Best Paper)
- **Kowalski–Mosteiro:** $O(n^{4+\epsilon}(\log^3 n)/\ell)$ rounds (ICALP 2019)
- **Kowalski–Mosteiro:** $\widetilde{O}(n^{2T(1+\epsilon)+3}/\ell)$ rounds (arXiv 2022)
- **Di Luna–V.:** $3n$ rounds (FOCS 2022) **This talk's focus**
- **Di Luna–V.:** $(\ell^2+\ell+1)Tn$ rounds (arXiv 2022) **A few words...**

Symbols:

- n: number of agents in the network (unknown)
- ℓ: number of Leaders (known; default: $\ell = 1$)
- T: connectivity parameter of the network (known; default: $T = 1$)
- N: upper bound on n (unknown, except in ICDCN 2014)

Today's talk

Theorem (Di Luna–V., FOCS 2022)

For $\ell = 1$ and $T = 1$, we have:

- *Stabilizing algorithm in $2n$ rounds (no termination).*
- *Terminating algorithm in $3n$ rounds.*
- *Lower bound of $2n$ rounds (for stabilization and termination).*

Local memory, local computation time, and message size are polynomial in n. Also works if the network is a multi-graph.

The theorem applies not only to the Counting Problem, but to *all* problems computable in anonymous (dynamic) networks.

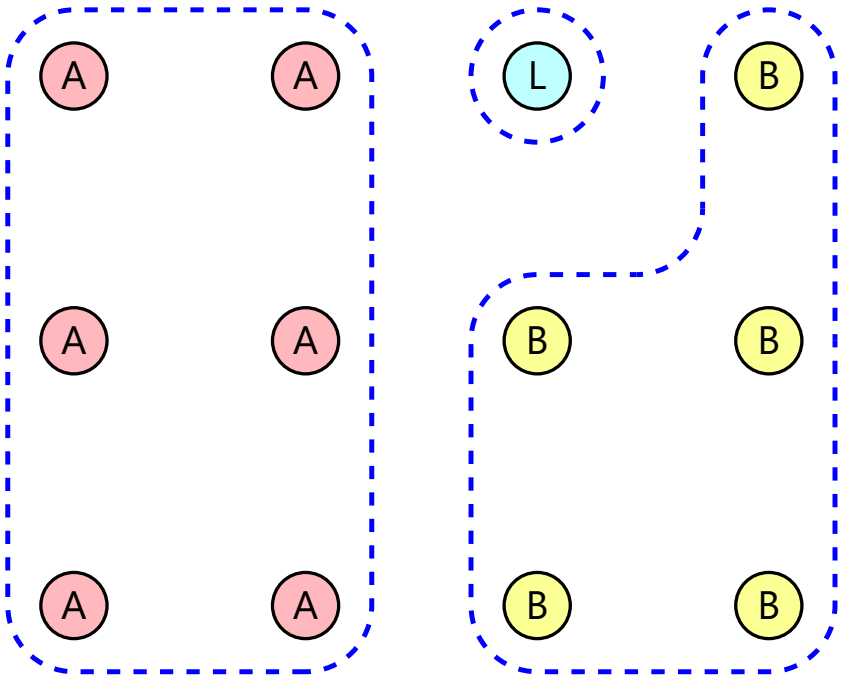
These are precisely the *multi-aggregate* functions f:

- Agent p outputs $f(x_p, \mu)$,
- where x_p is the input of agent p,
- and μ is the multi-set of all inputs.

Computability

General computation

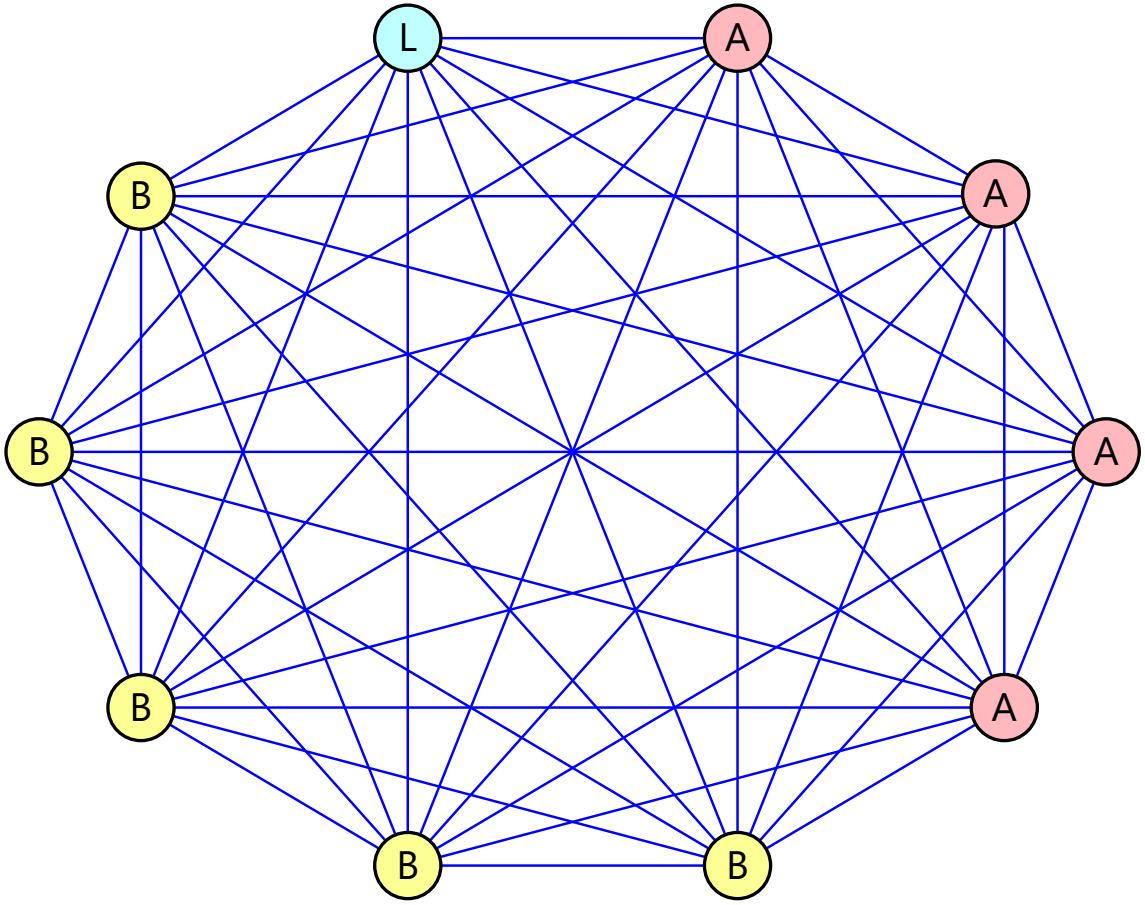
In general, we may assume that each agent has an *input* and has to compute an *output* depending on the entire network's inputs.



Agents with the same input are still indistinguishable (anonymous).

General computation

If the network is the complete graph at every round, all agents with the same input will always have the same internal state.



Thus, an agent's output can only depend on its input and the *number* of agents having each input.

Completeness of the Generalized Counting Problem

Thus, only the *multi-aggregate* functions can be computed.

Observation

If a function is computable in an anonymous dynamic network (with a unique Leader), it must be a multi-aggregate function.

Examples: The average, maximum, minimum, sum, mode, variance, and most statistical functions are (multi-)aggregate.

Generalized Counting Problem: Eventually, all agents must know how many agents have each input.

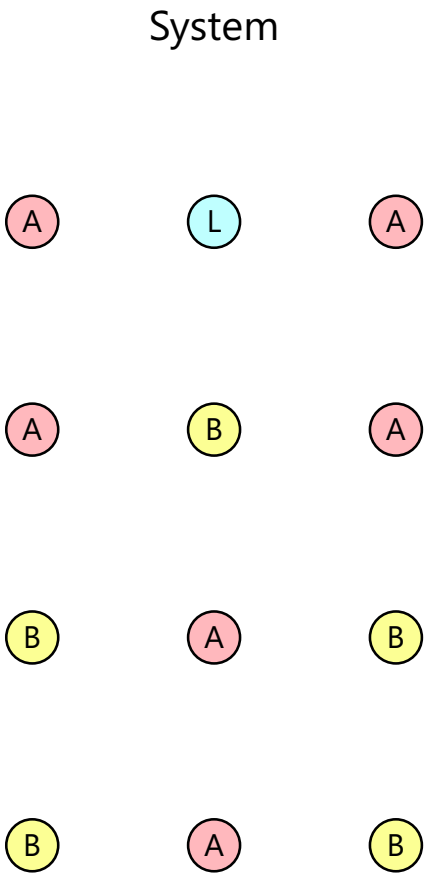
Observation

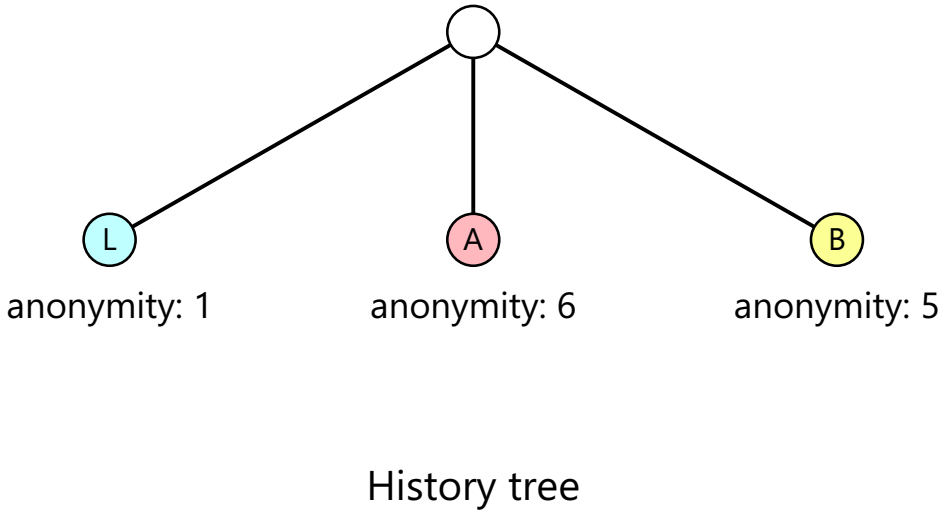
If the Generalized Counting Problem is solvable in $f(n)$ rounds, then every multi-aggregate function is computable in $f(n)$ rounds.

History Trees

History trees

We introduce the *history tree* as our main tool of investigation.

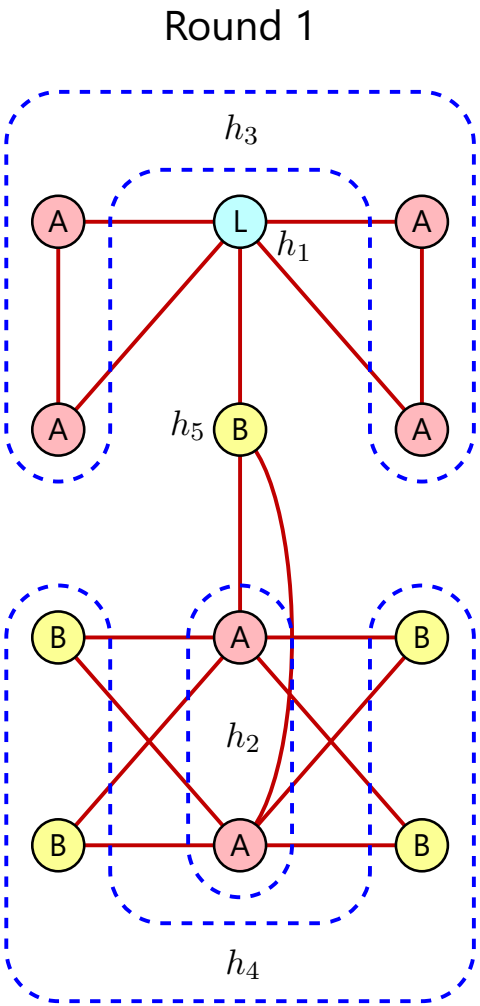


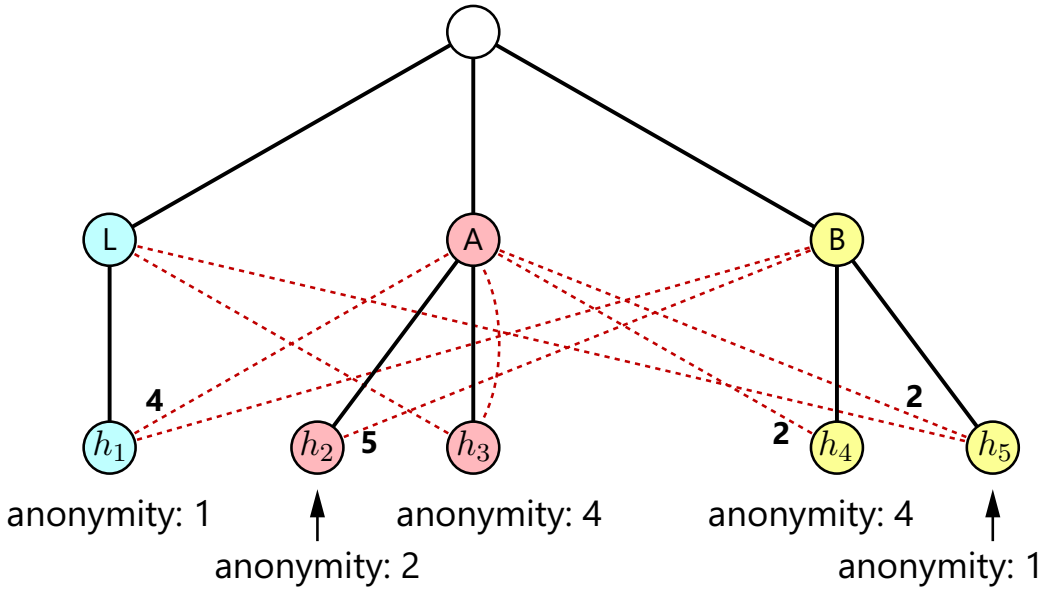


History tree

History trees

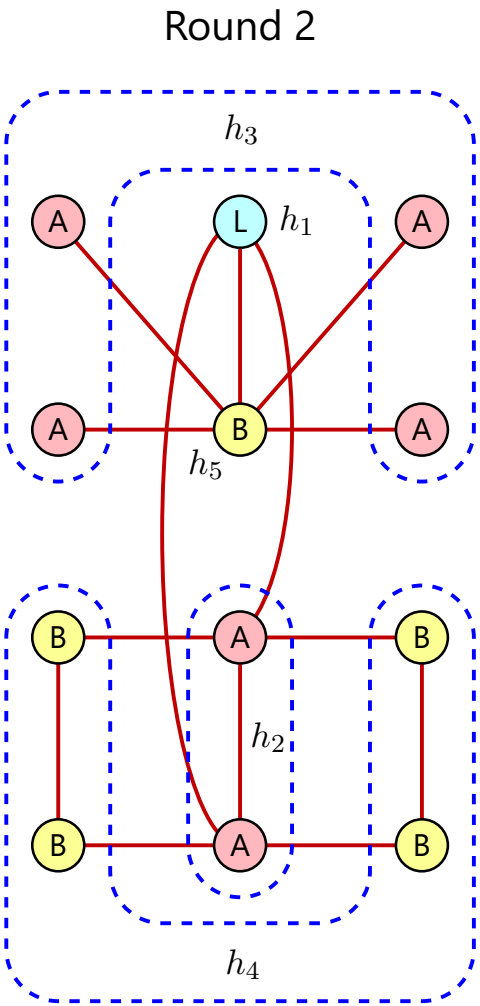
We introduce the *history tree* as our main tool of investigation.

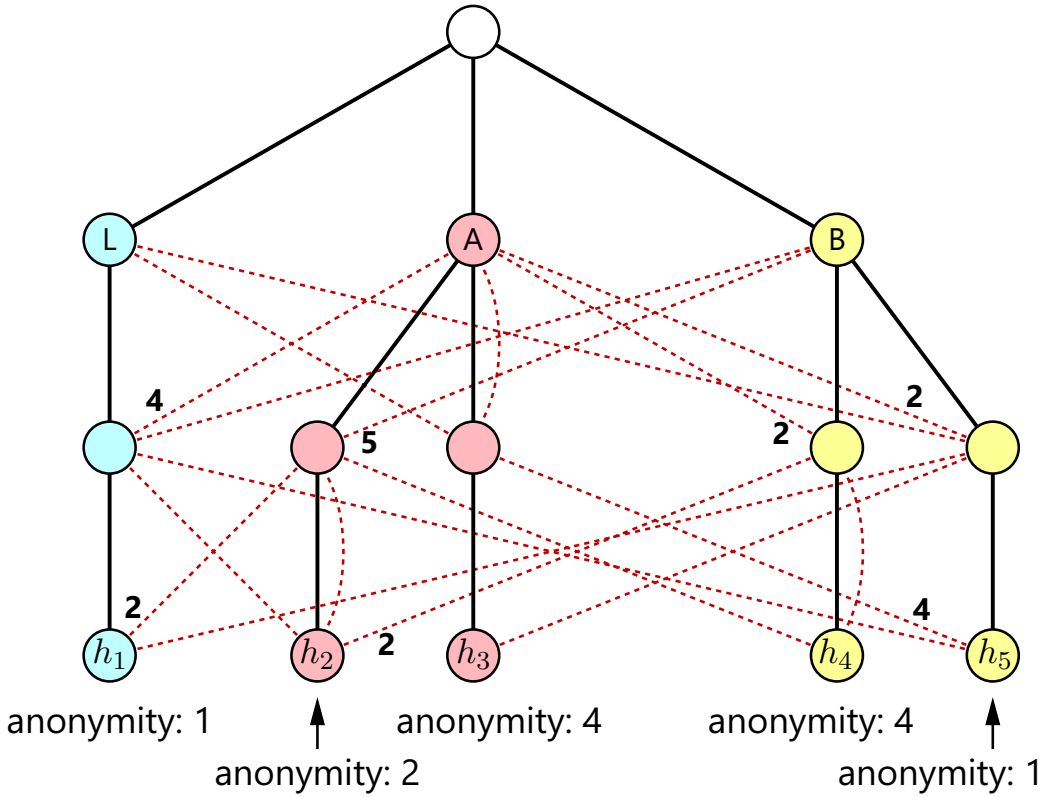




History trees

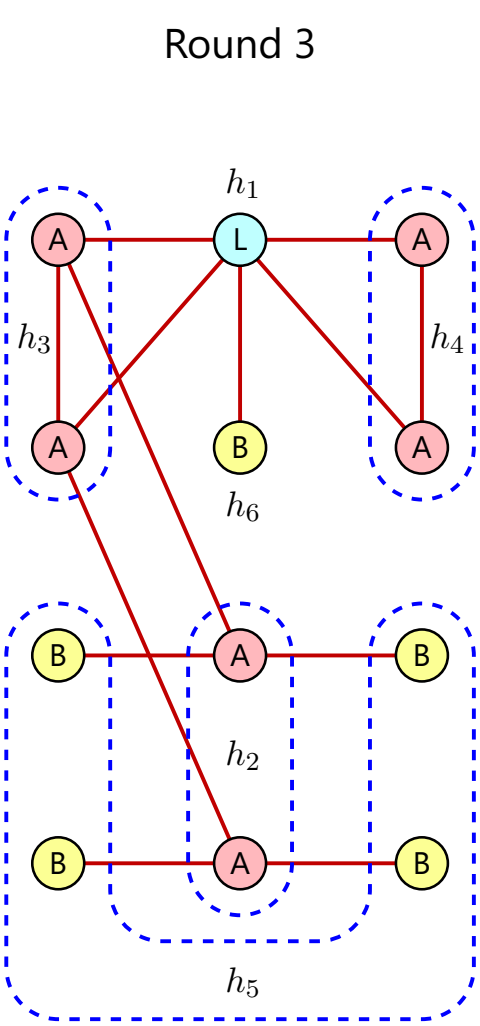
We introduce the *history tree* as our main tool of investigation.

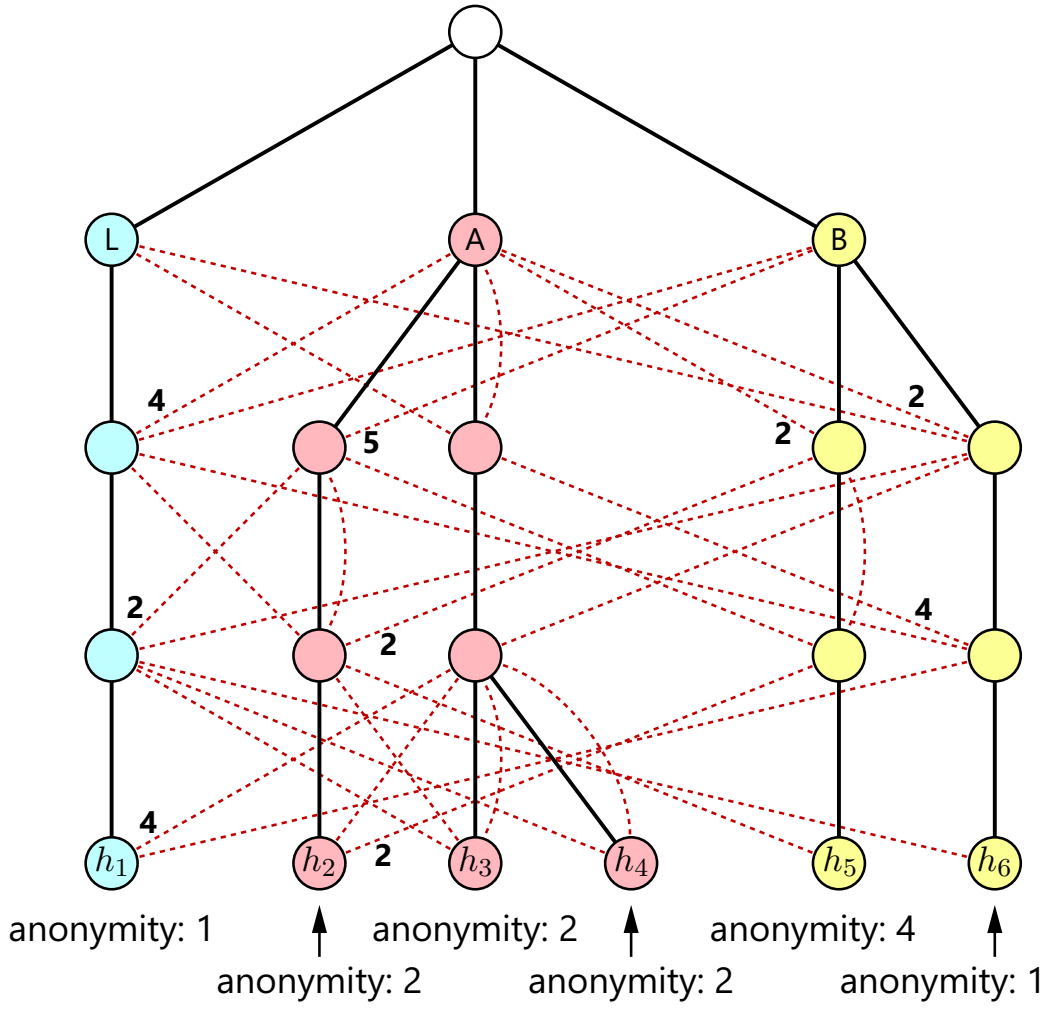




History trees

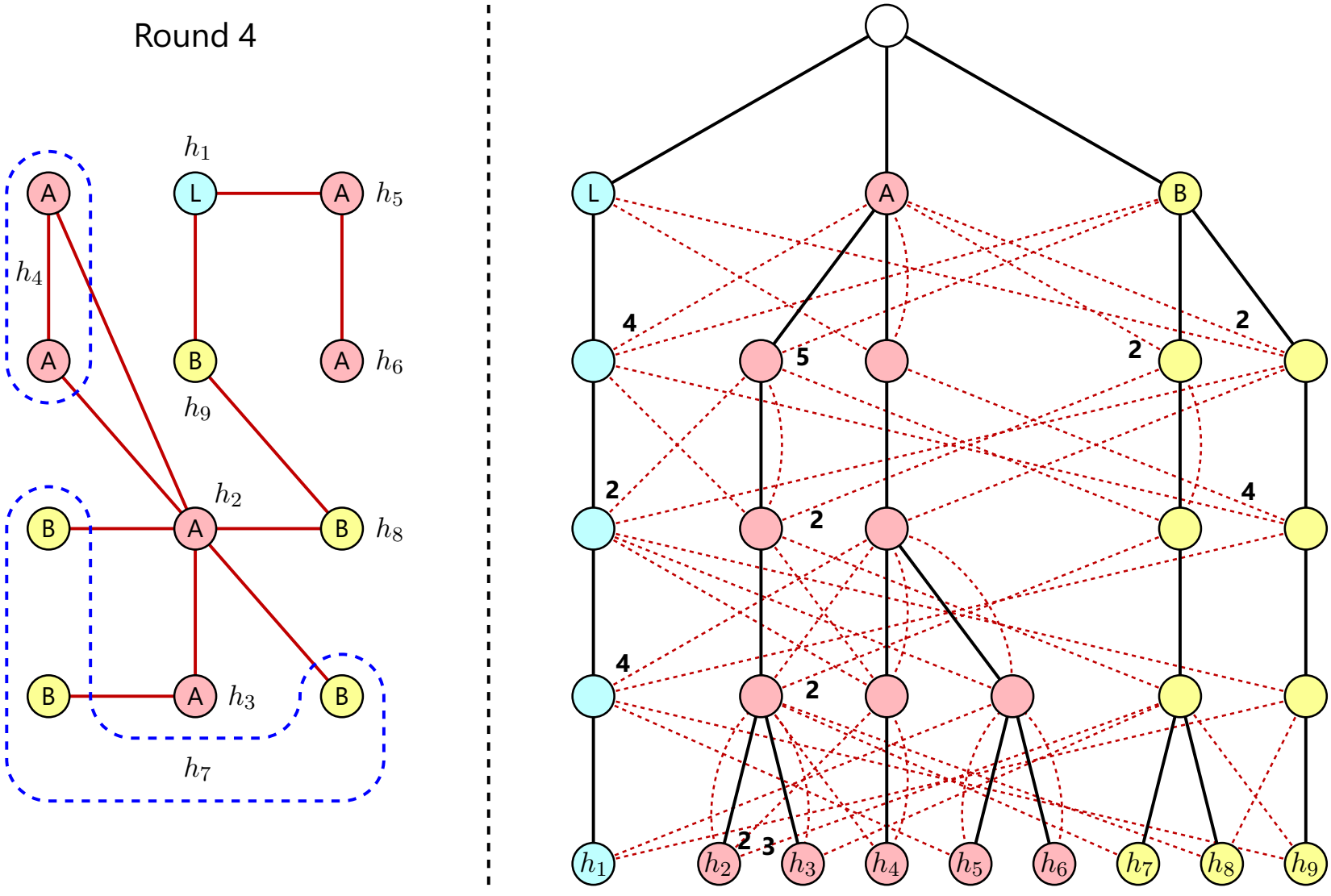
We introduce the *history tree* as our main tool of investigation.





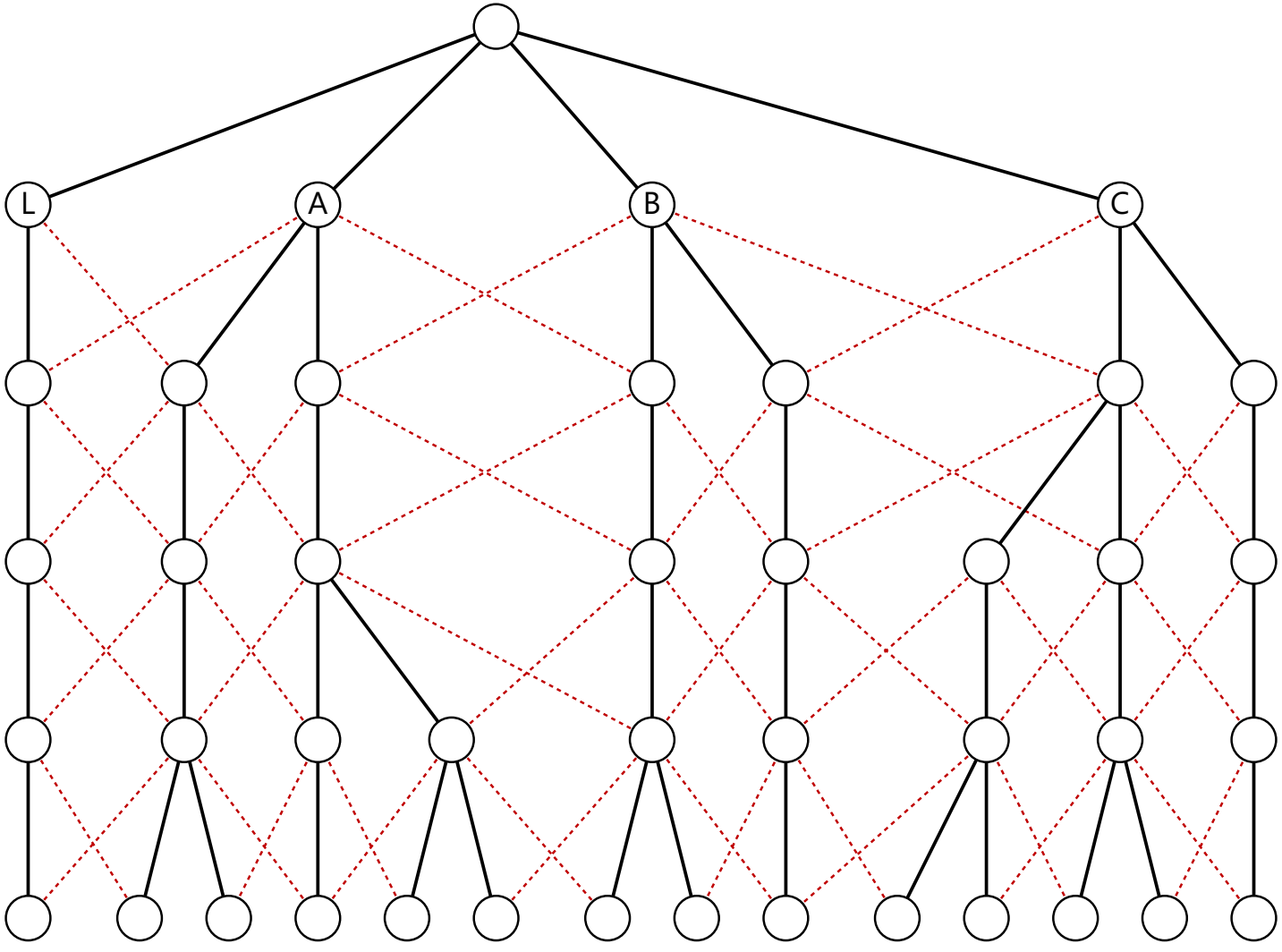
History trees

We introduce the *history tree* as our main tool of investigation.



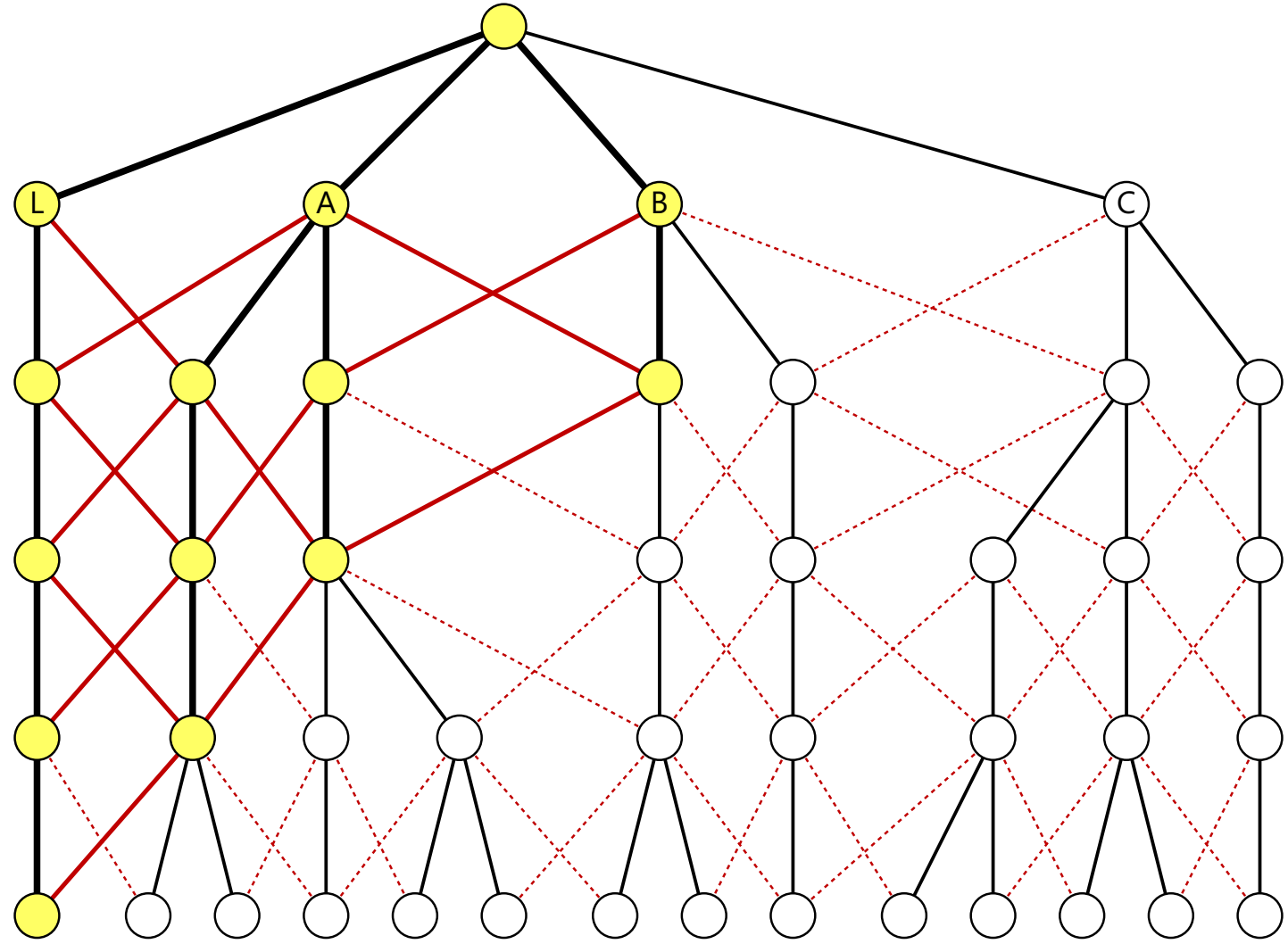
View of a history tree

At any point in time, an agent only has a *view* of the history tree.



View of a history tree

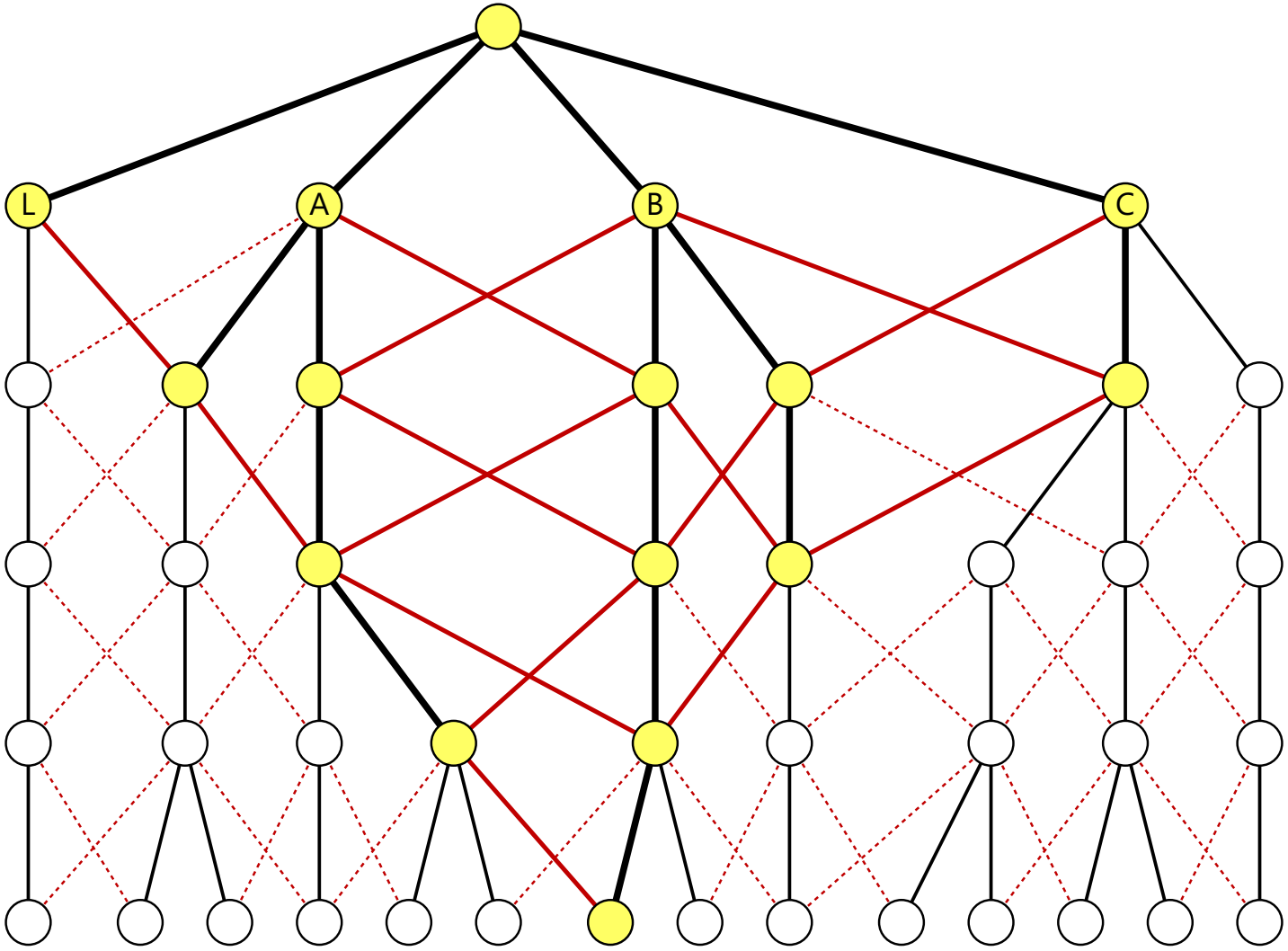
At any point in time, an agent only has a *view* of the history tree.



view

View of a history tree

At any point in time, an agent only has a *view* of the history tree.



view

Views as internal states and messages

An agent's view summarizes its whole *history* up to some round.

Observation

Without loss of generality, we may assume that an agent's internal state coincides with its view of the history tree.

Observation

Without loss of generality, we may assume that an agent broadcasts its own internal state at every round.

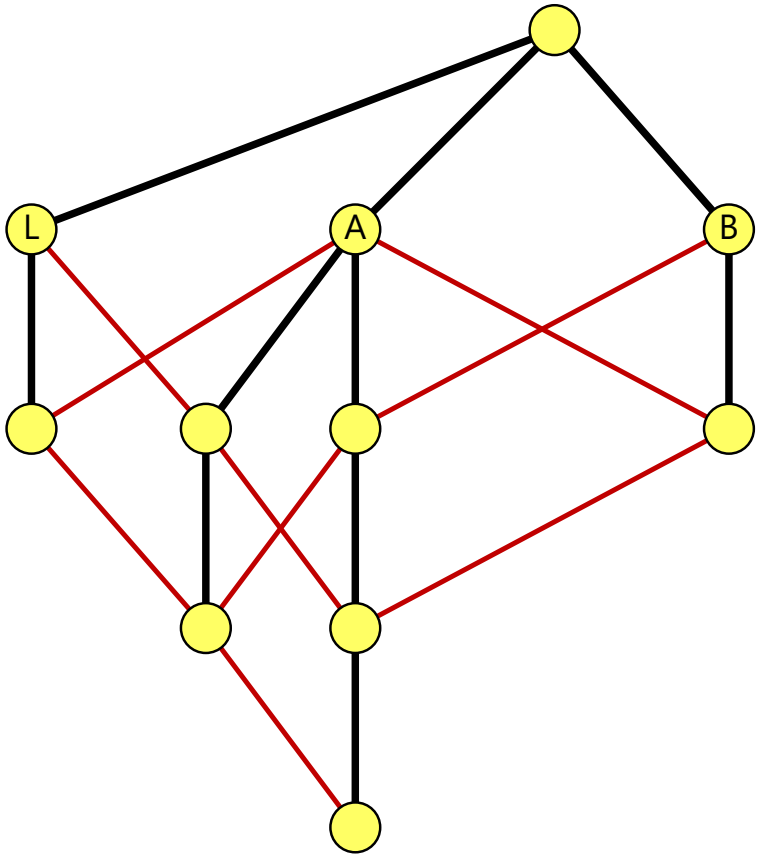
This is good because, at round i, the size of a view is only $O(i^3)$.

Observation

If a problem is solvable in a polynomial number of rounds, it can be solved by using a polynomial amount of local memory and sending messages of polynomial size.

Updating the view

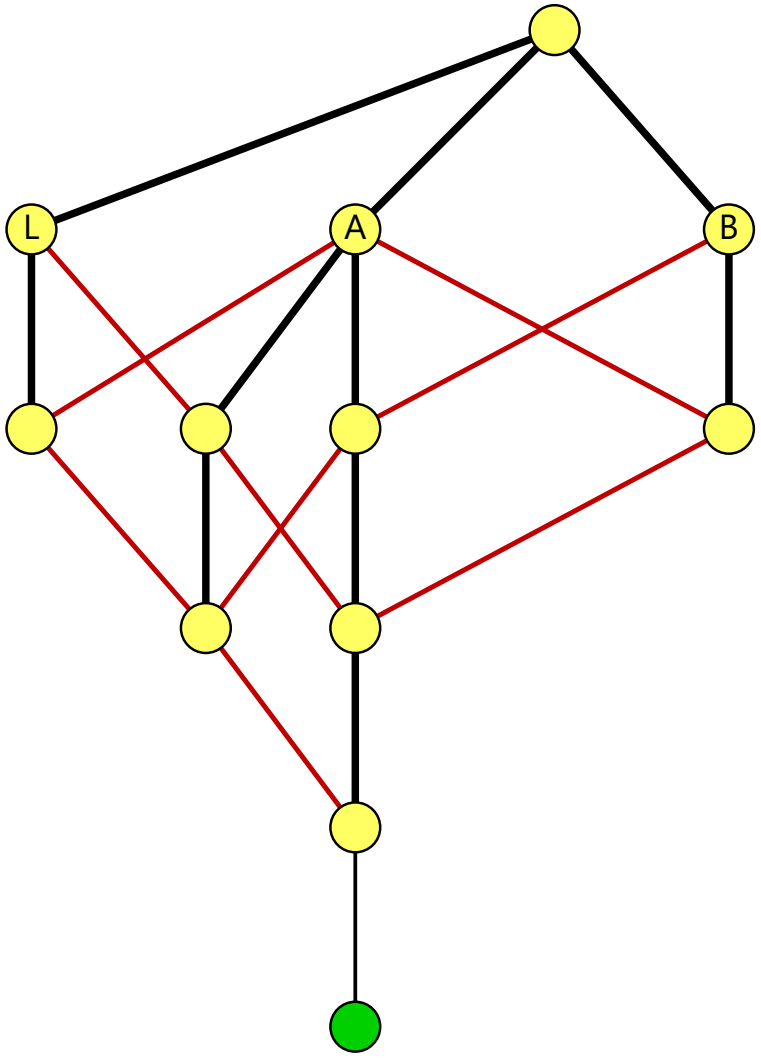
An agent updates its internal state by *merging* its view with the views it receives from its neighbors.



view

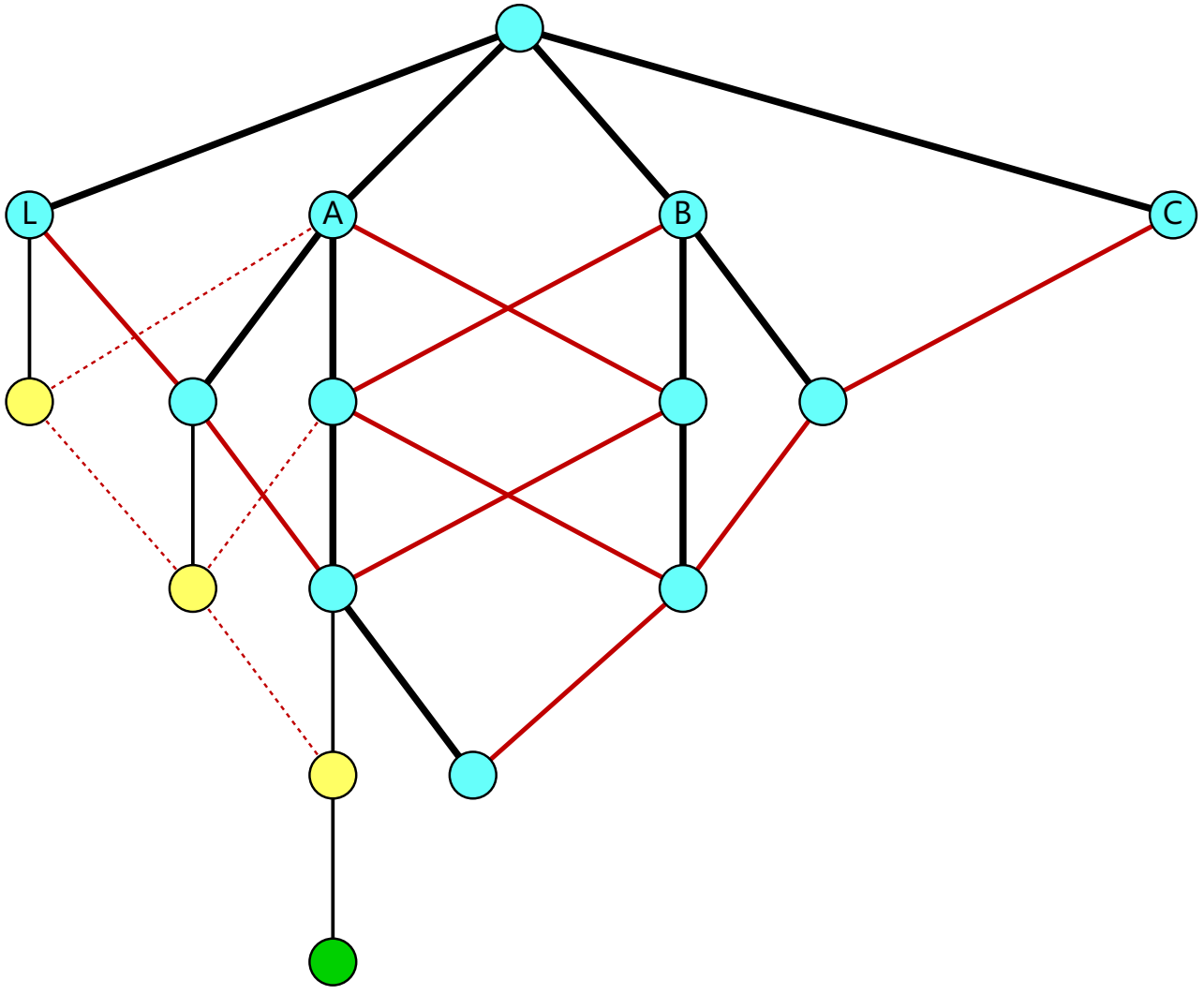
Updating the view

An agent updates its internal state by *merging* its view with the views it receives from its neighbors.



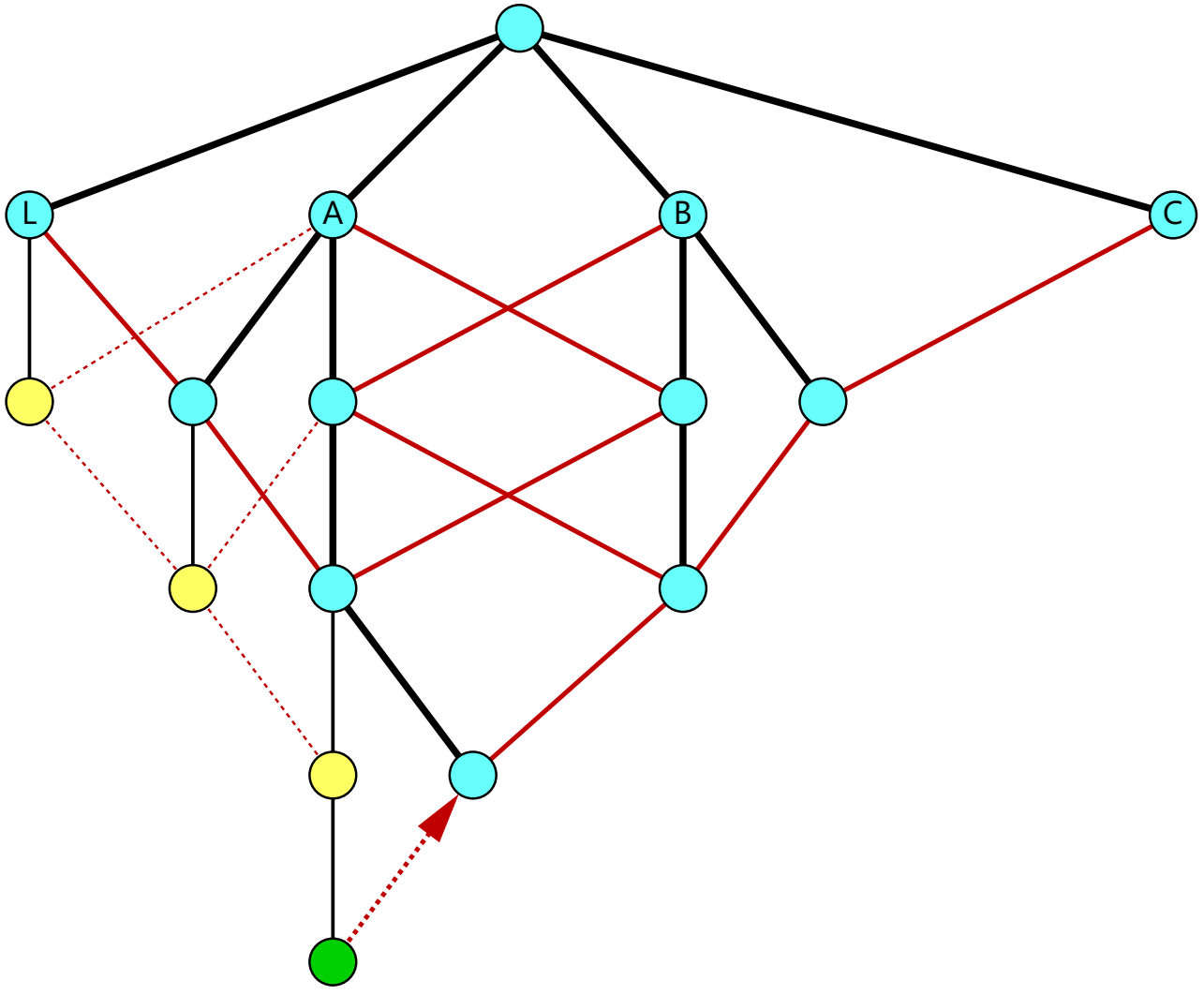
Updating the view

An agent updates its internal state by *merging* its view with the views it receives from its neighbors.



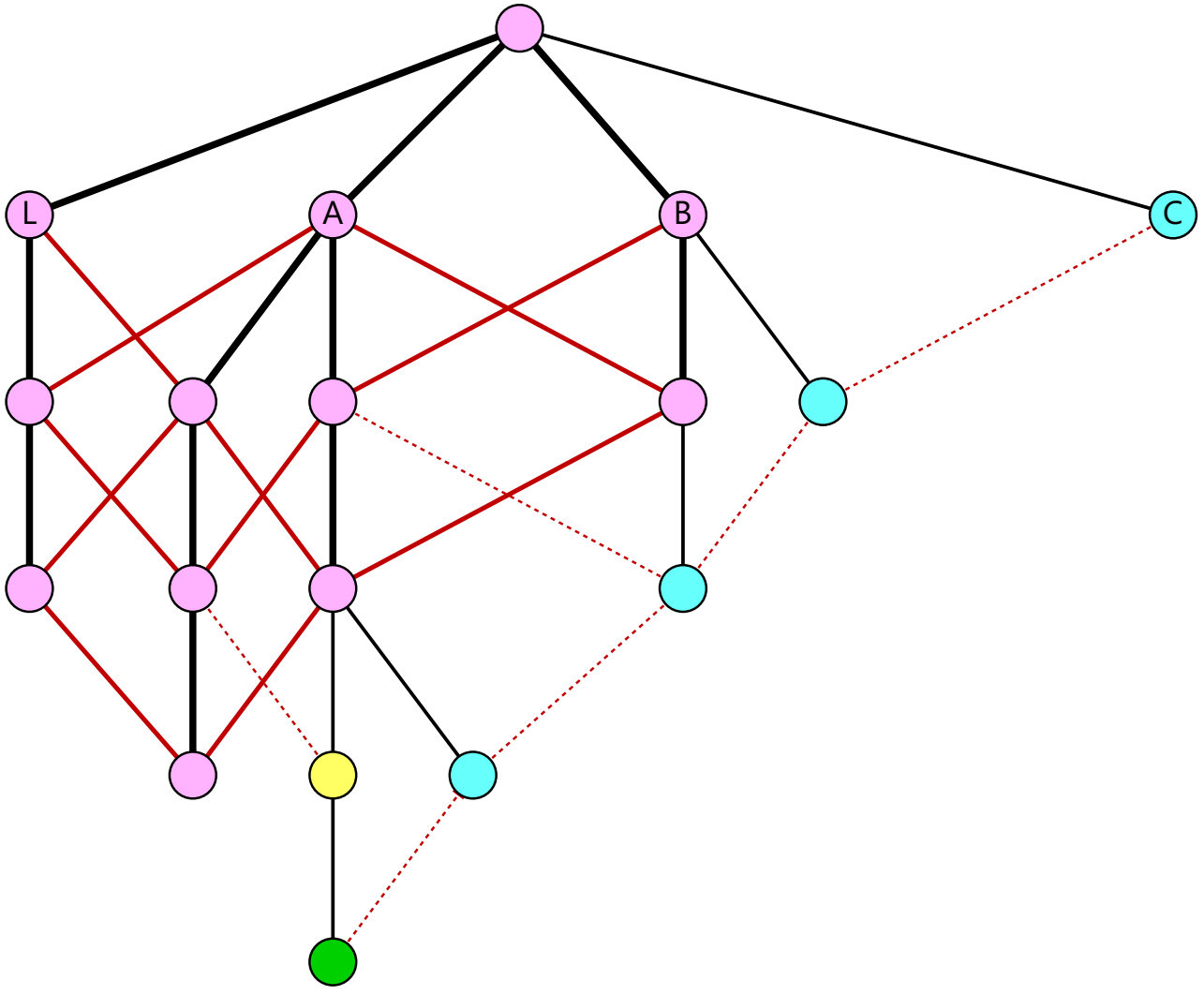
Updating the view

An agent updates its internal state by *merging* its view with the views it receives from its neighbors.



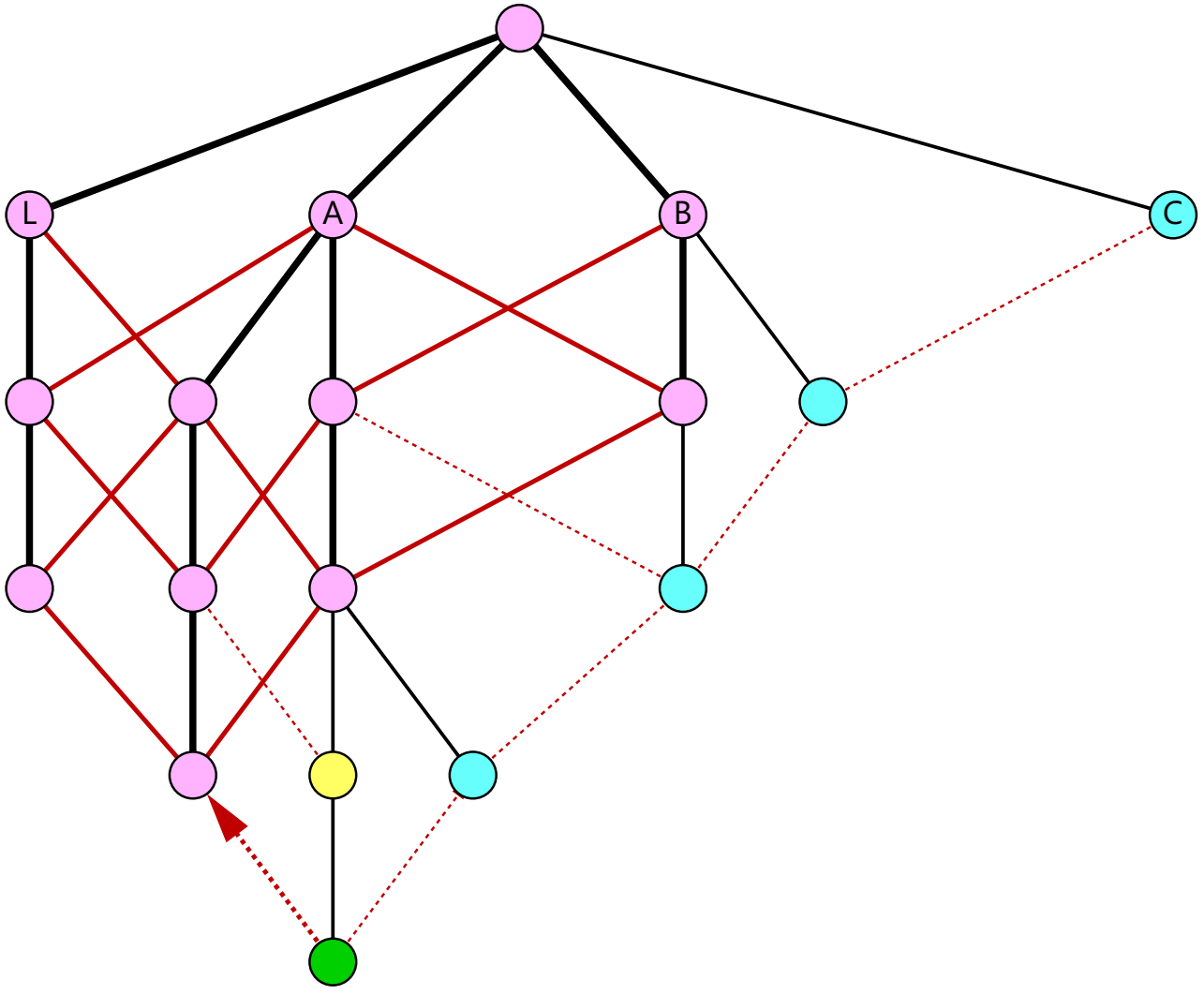
Updating the view

An agent updates its internal state by *merging* its view with the views it receives from its neighbors.



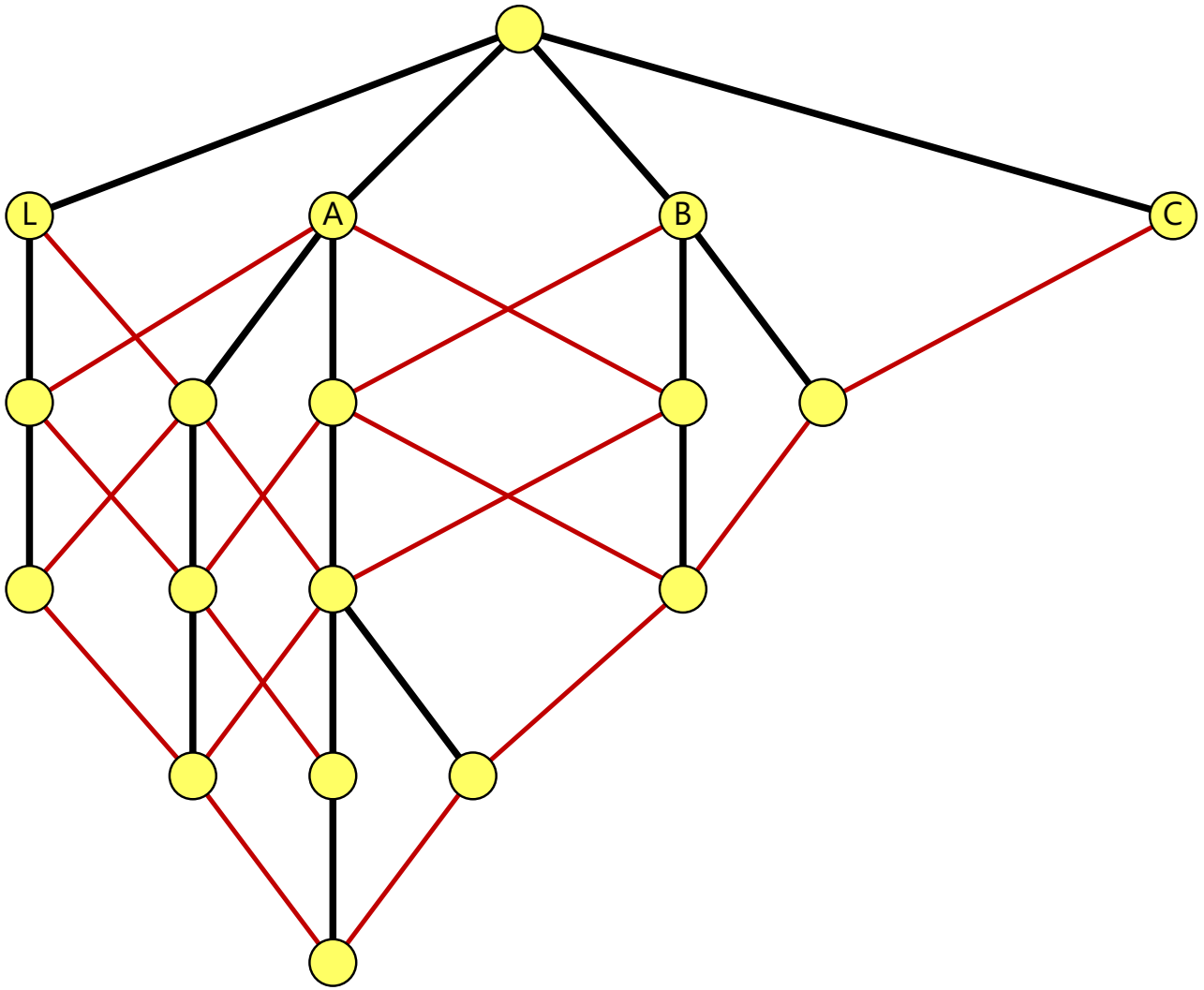
Updating the view

An agent updates its internal state by *merging* its view with the views it receives from its neighbors.



Updating the view

An agent updates its internal state by *merging* its view with the views it receives from its neighbors.



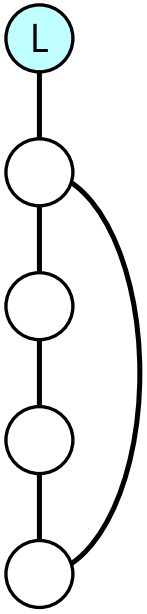
Lower Bound on Counting

Lower bound

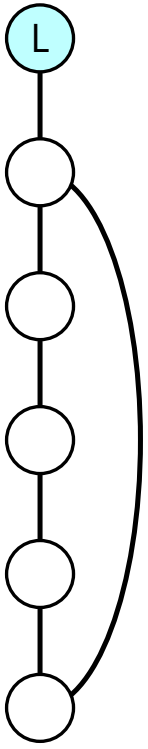
Theorem

The Counting Problem is not solvable in less than $2n - 3$ rounds.

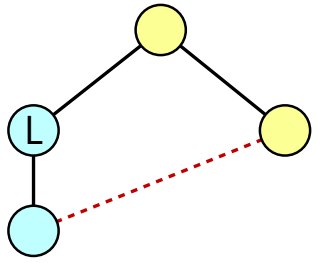
System 1



System 2



Leaders' view



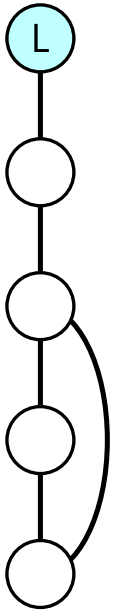
Round 1

Lower bound

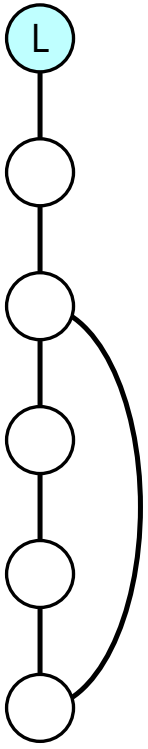
Theorem

The Counting Problem is not solvable in less than $2n - 3$ rounds.

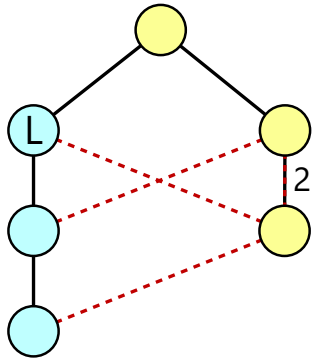
System 1



System 2



Leaders' view



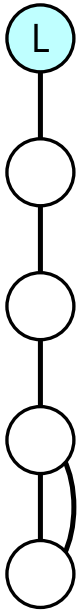
Round 2

Lower bound

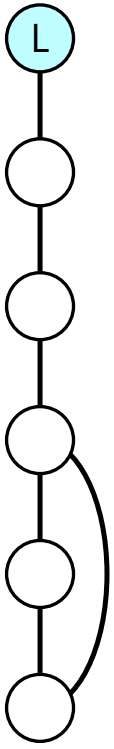
Theorem

The Counting Problem is not solvable in less than $2n - 3$ rounds.

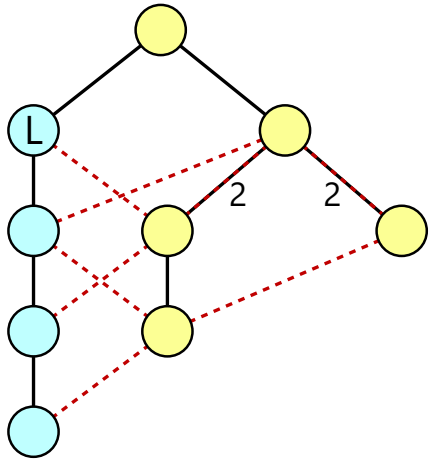
System 1



System 2



Leaders' view



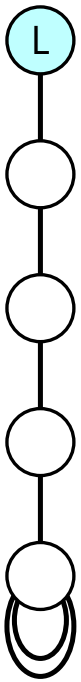
Round 3

Lower bound

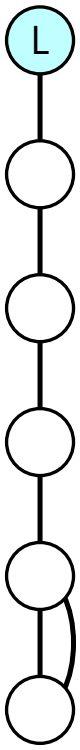
Theorem

The Counting Problem is not solvable in less than $2n - 3$ rounds.

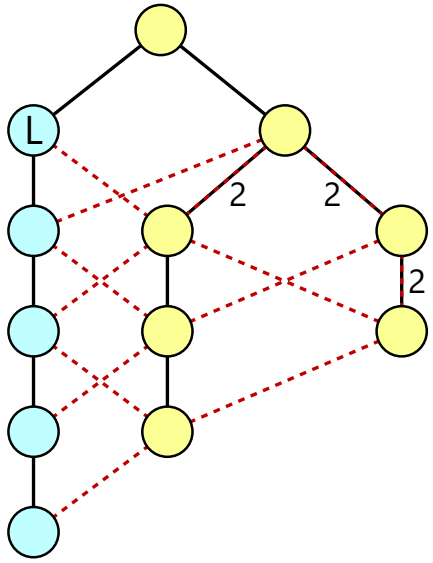
System 1



System 2



Leaders' view



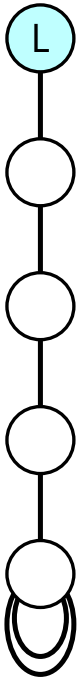
Round 4

Lower bound

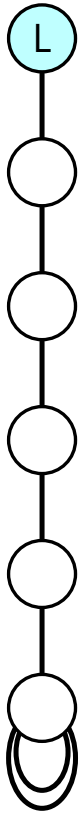
Theorem

The Counting Problem is not solvable in less than $2n - 3$ rounds.

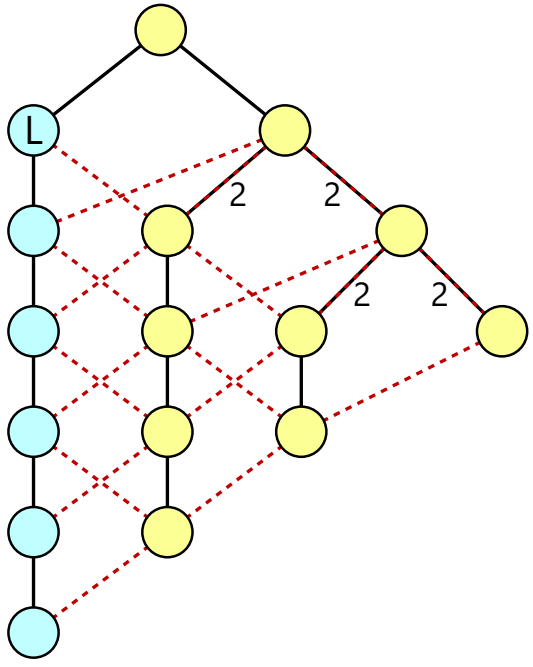
System 1



System 2



Leaders' view



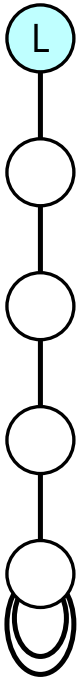
Round 5

Lower bound

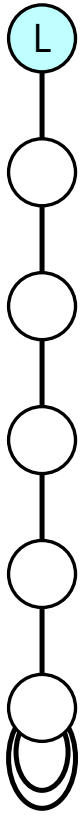
Theorem

The Counting Problem is not solvable in less than $2n - 3$ rounds.

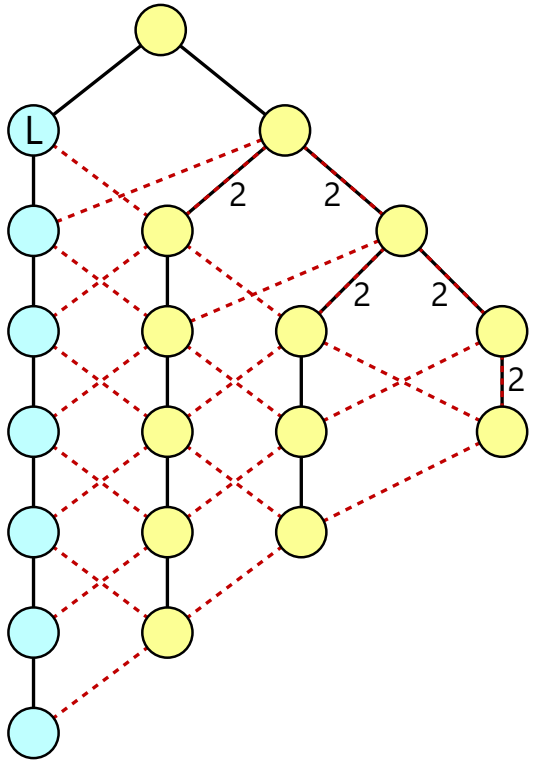
System 1



System 2



Leaders' view



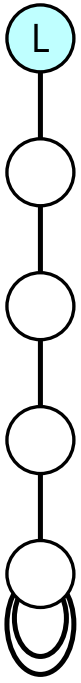
Round 6

Lower bound

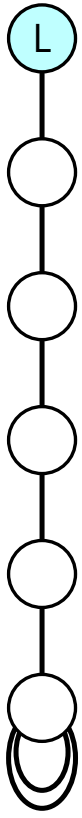
Theorem

The Counting Problem is not solvable in less than $2n - 3$ rounds.

System 1

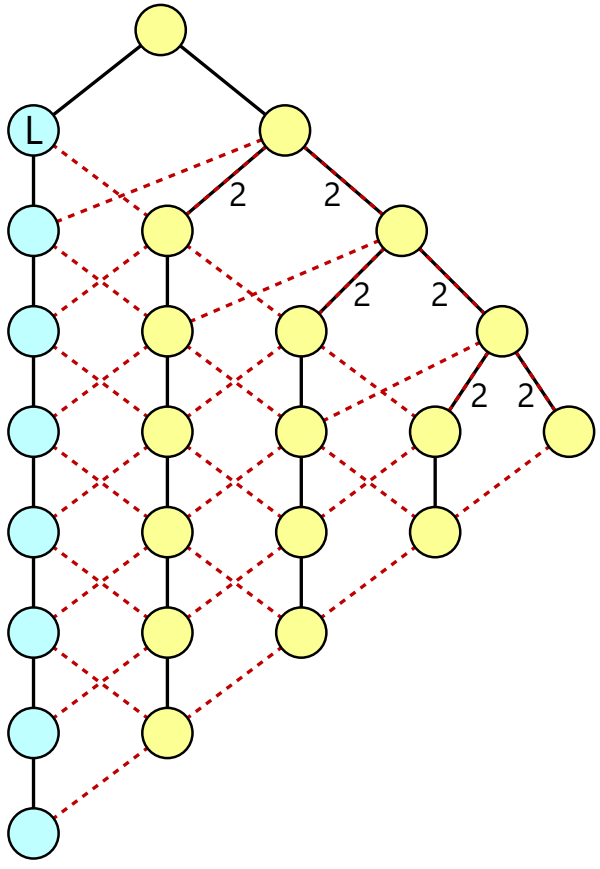


System 2



Round 7

Leaders' view

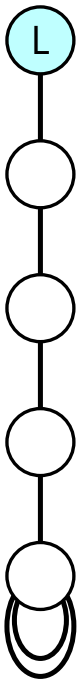


Lower bound

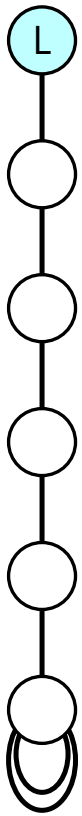
Theorem

The Counting Problem is not solvable in less than $2n - 3$ rounds.

System 1

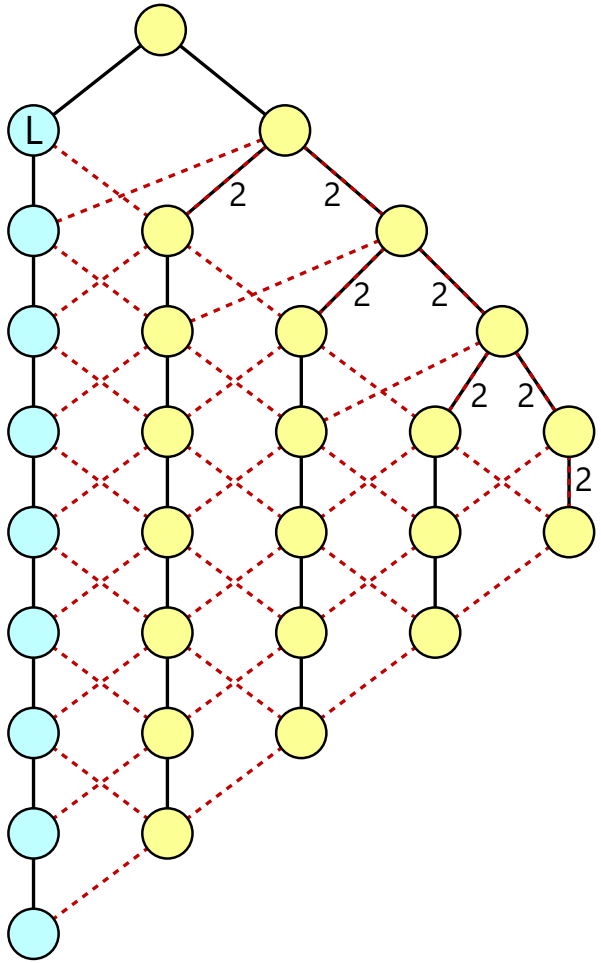


System 2



Round 8

Leaders' view

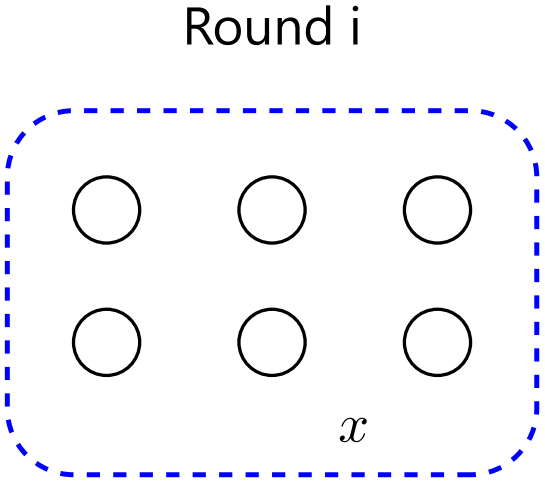


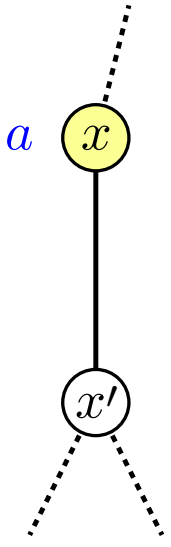
Stabilizing Algorithm

Computing anonymities

Suppose we know the anonymity of a node x in the history tree.

If x has only one child x', then x' must have the same anonymity.

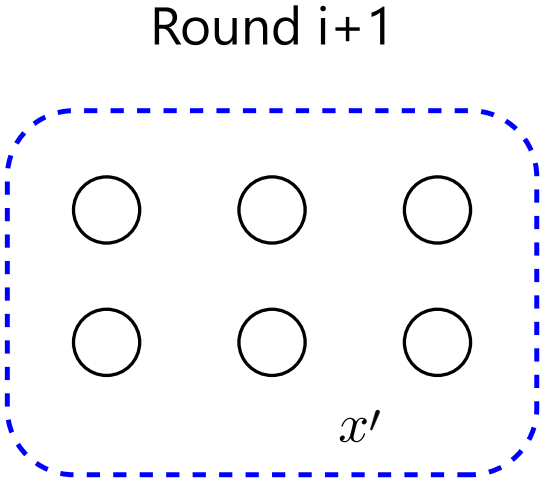


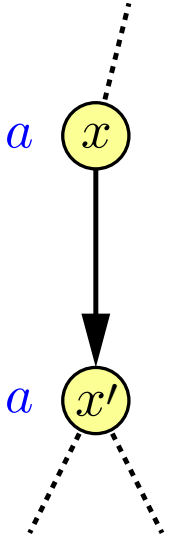


Computing anonymities

Suppose we know the anonymity of a node x in the history tree.

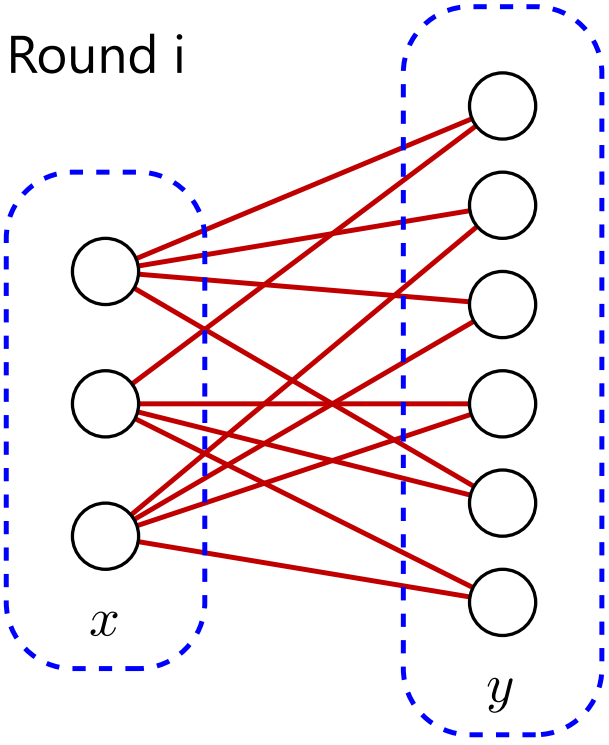
If x has only one child x', then x' must have the same anonymity.

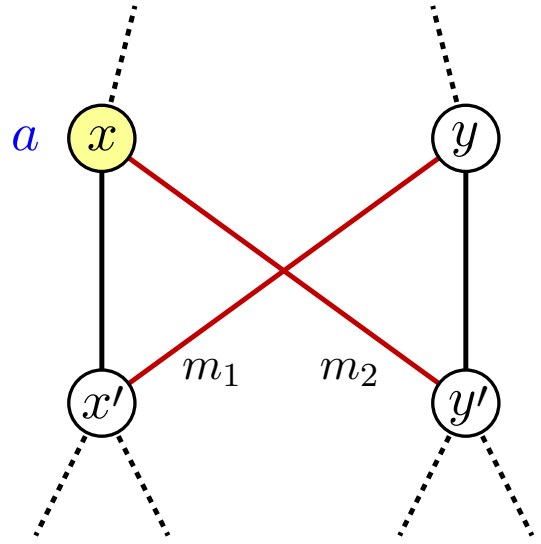




Computing anonymities

Suppose we know the anonymity of a node x with a single child x'.

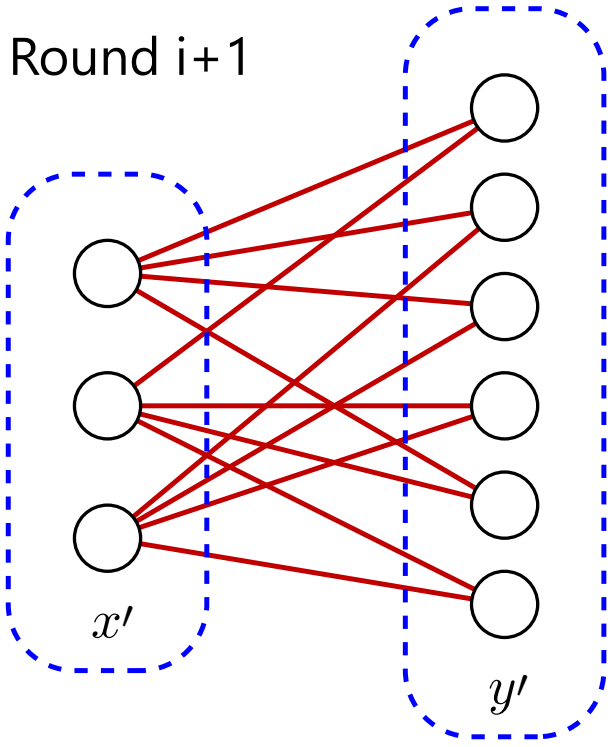


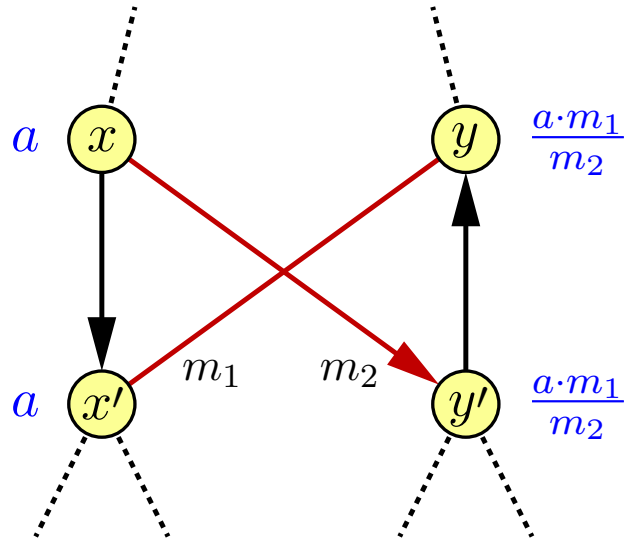


If the agents represented by x have observed agents whose corresponding node y has only one child y', then we can compute the anonymity of y and y', as well.

Computing anonymities

Suppose we know the anonymity of a node x with a single child x'.

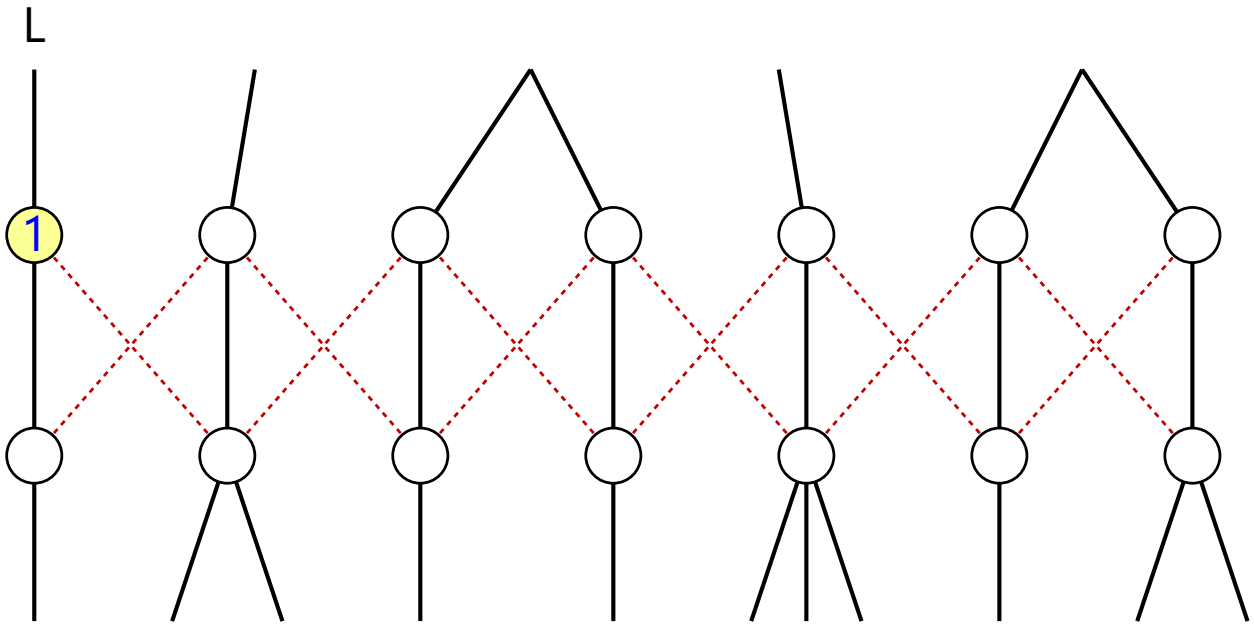




If the agents represented by x have observed agents whose corresponding node y has only one child y', then we can compute the anonymity of y and y', as well.

Stabilizing algorithm

If all nodes in a level have only one child, we can compute the anonymity of all of them (because the network is connected).

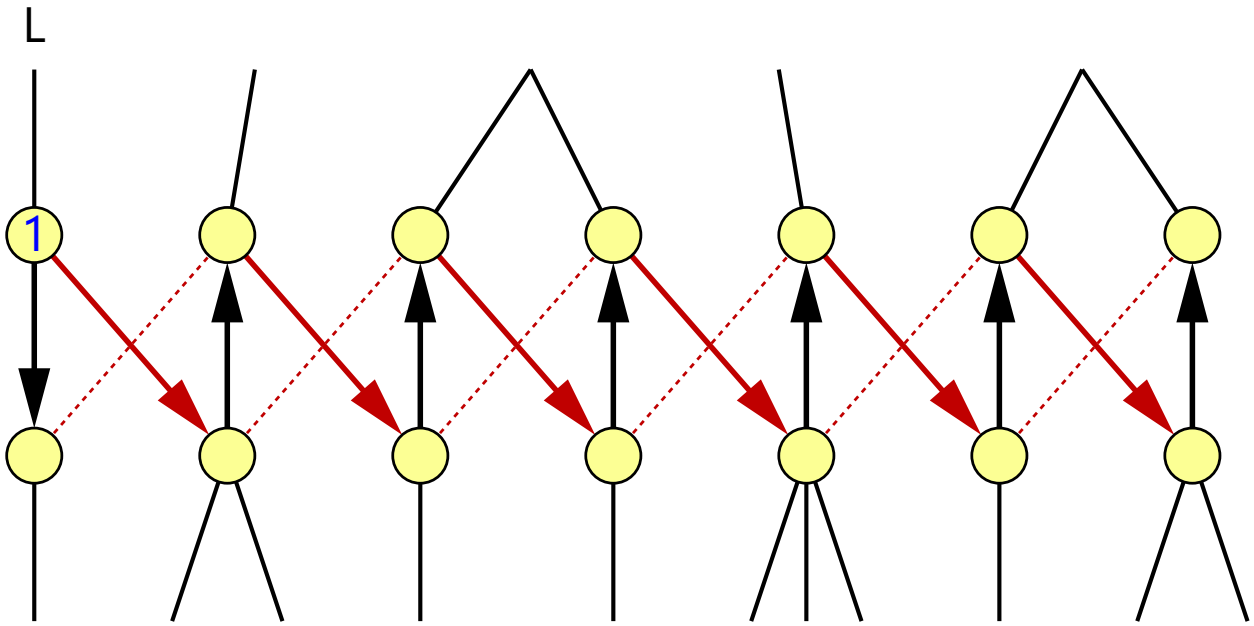


Since there are n agents, the tree can branch at most $n-1$ times.

Thus, among the first $n-1$ levels, there must be a level where no node branches. In this level, we can compute all anonymities.

Stabilizing algorithm

If all nodes in a level have only one child, we can compute the anonymity of all of them (because the network is connected).



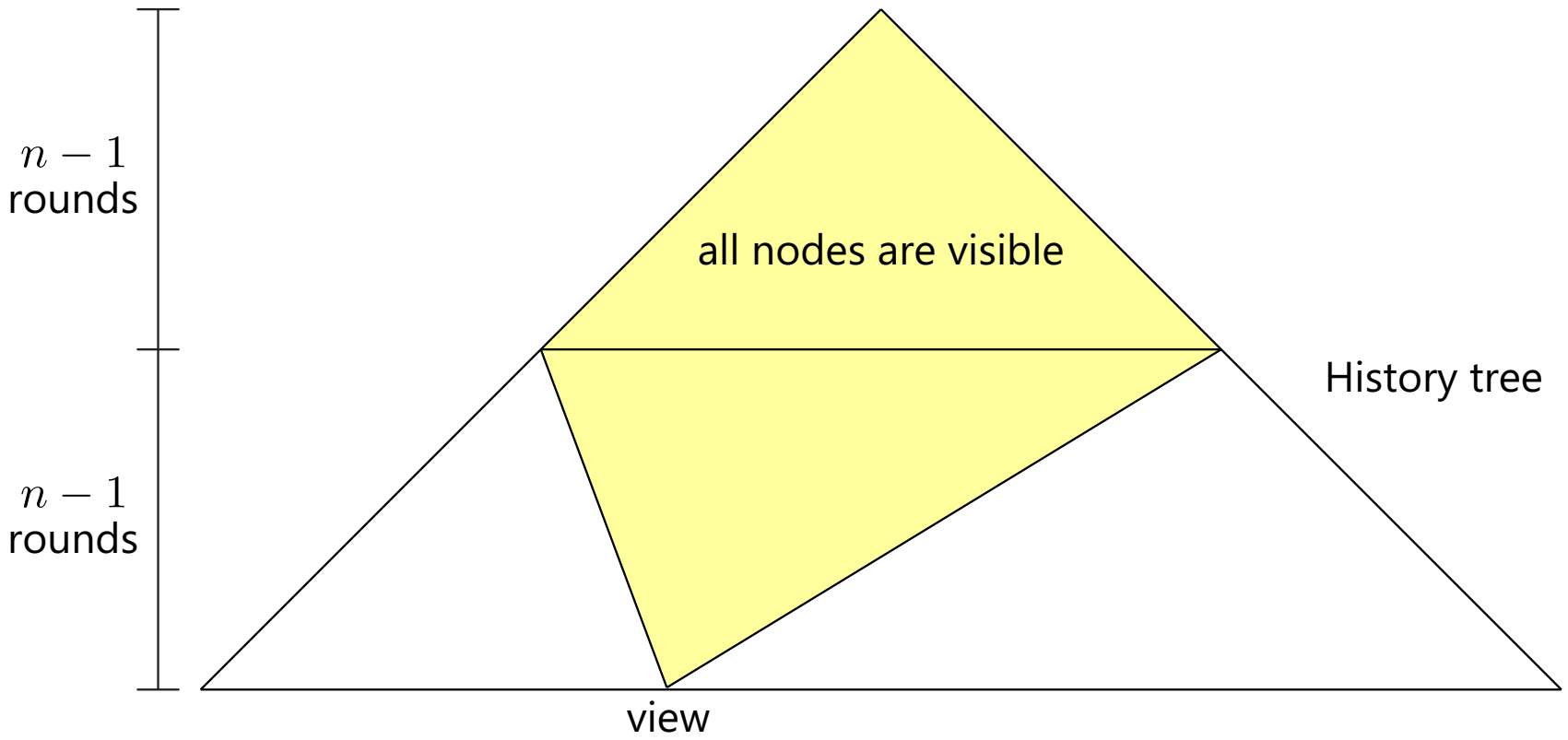
Since there are n agents, the tree can branch at most $n-1$ times.

Thus, among the first $n-1$ levels, there must be a level where no node branches. In this level, we can compute all anonymities.

Stabilizing algorithm

Theorem

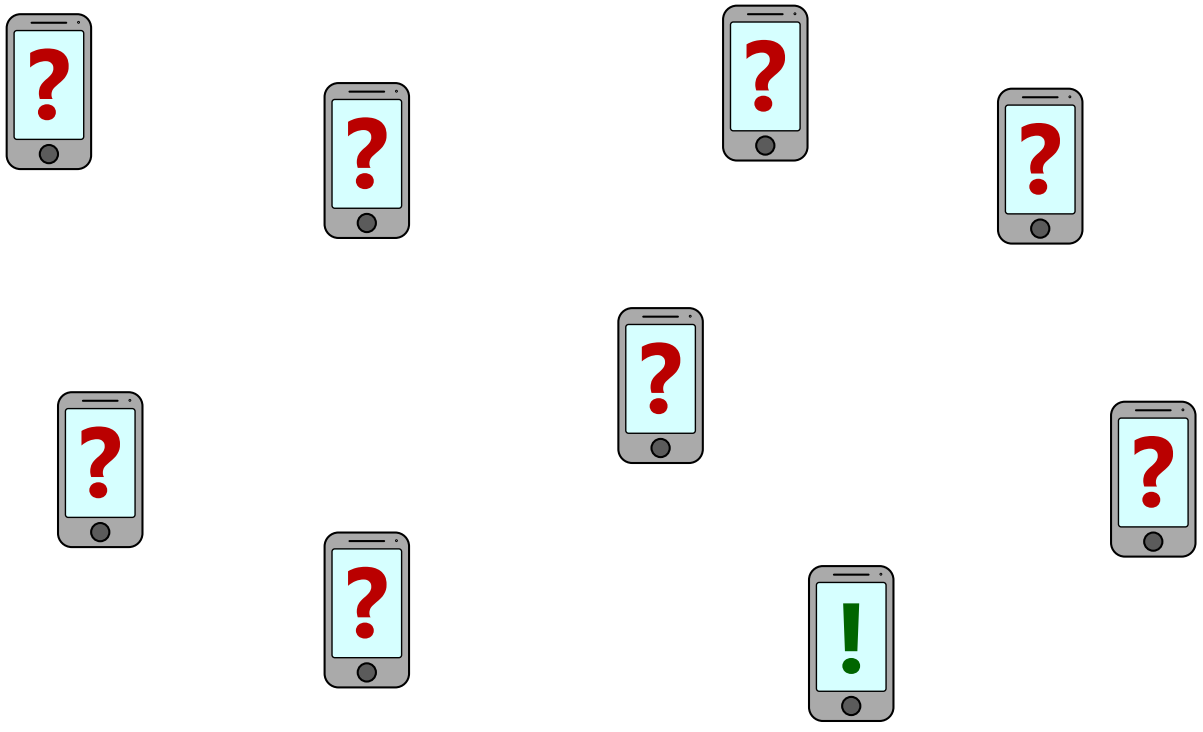
The Generalized Counting Problem can be stably solved in $2n-2$ rounds (without explicit termination).



Note that, after $2n-2$ rounds, all nodes in the first $n-1$ levels of the history tree are in the views of all agents.

Propagation of information

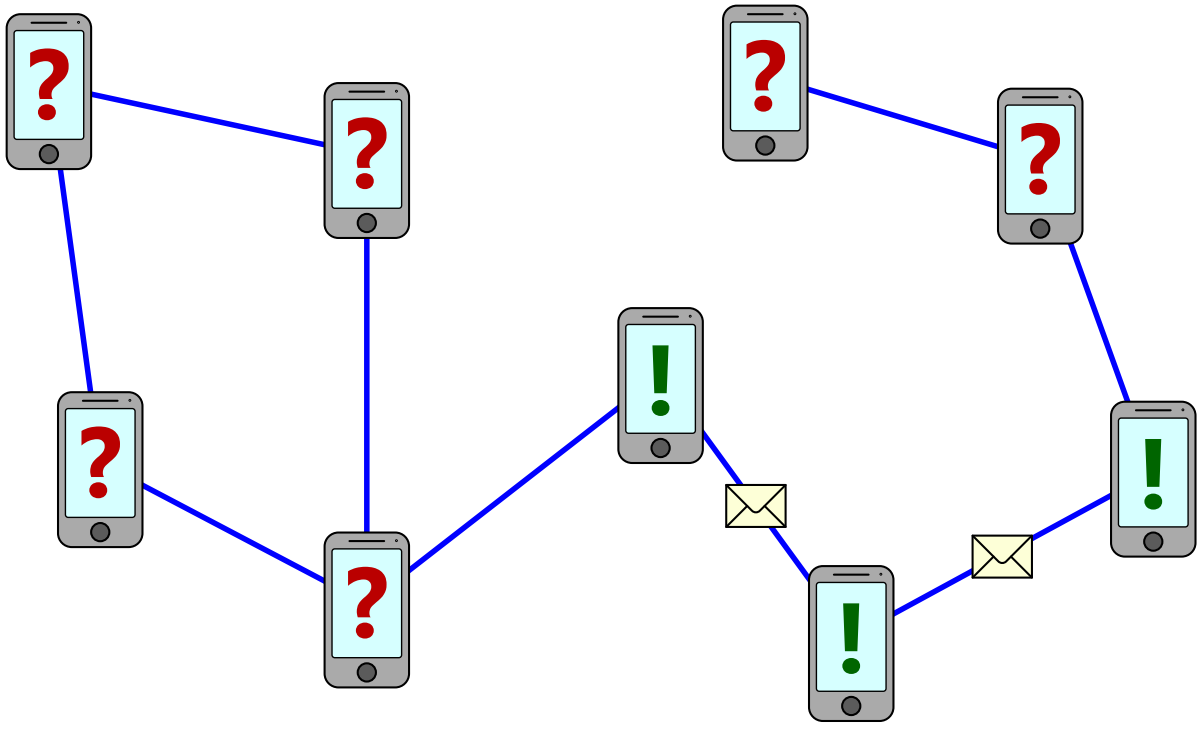
If the network is connected at all rounds, every news reaches every agent in at most $n - 1$ rounds.



Hence, whenever two agents interact, all agents will know it within $n - 1$ rounds (and it will show in their views of the history tree).

Propagation of information

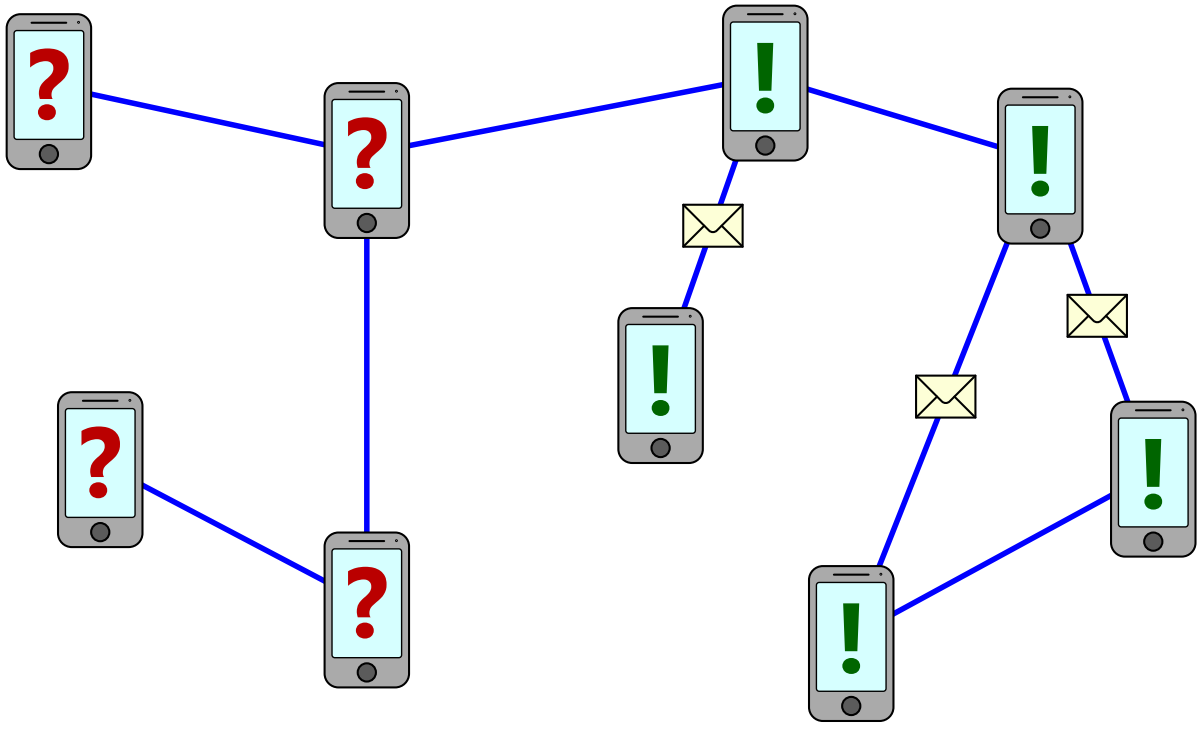
If the network is connected at all rounds, every news reaches every agent in at most $n - 1$ rounds.



Hence, whenever two agents interact, all agents will know it within $n - 1$ rounds (and it will show in their views of the history tree).

Propagation of information

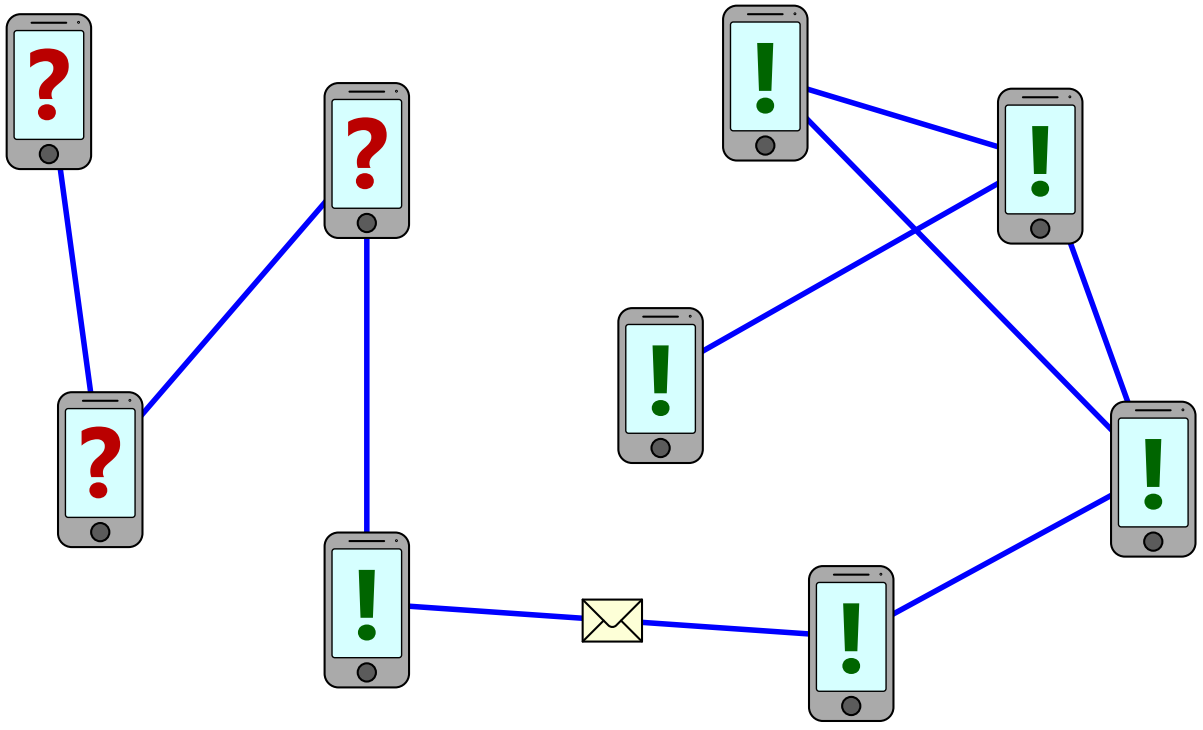
If the network is connected at all rounds, every news reaches every agent in at most $n - 1$ rounds.



Hence, whenever two agents interact, all agents will know it within $n - 1$ rounds (and it will show in their views of the history tree).

Propagation of information

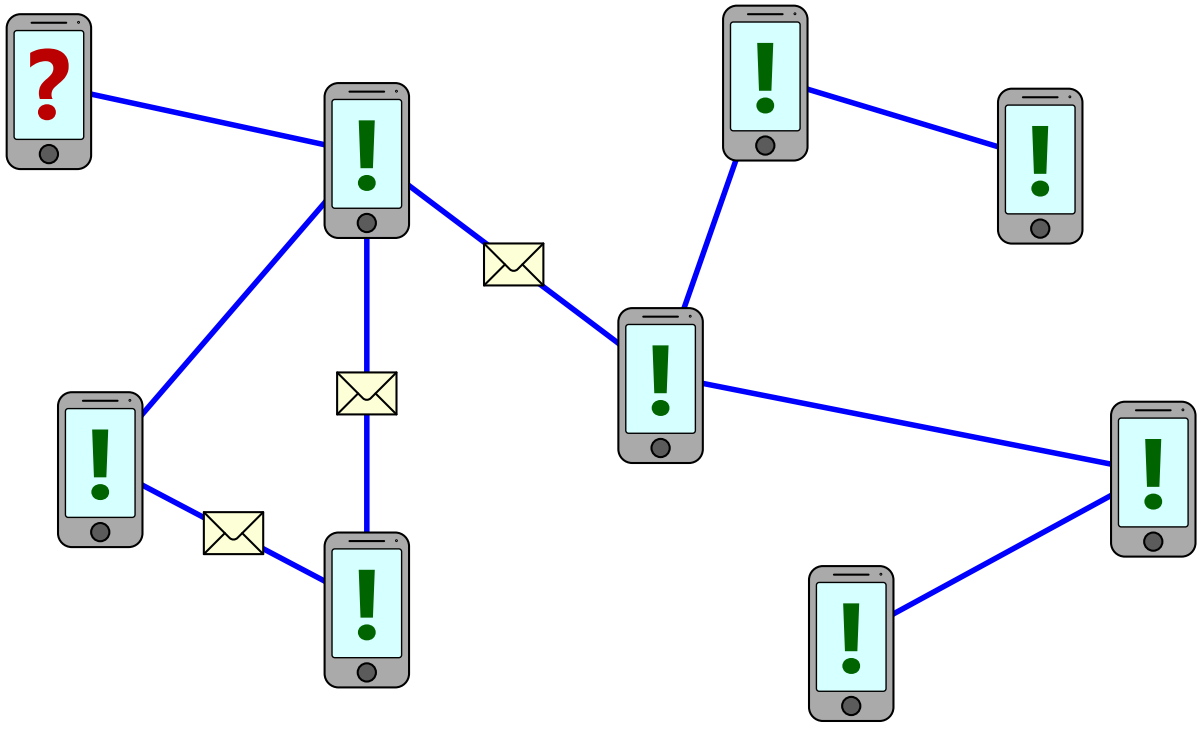
If the network is connected at all rounds, every news reaches every agent in at most $n - 1$ rounds.



Hence, whenever two agents interact, all agents will know it within $n - 1$ rounds (and it will show in their views of the history tree).

Propagation of information

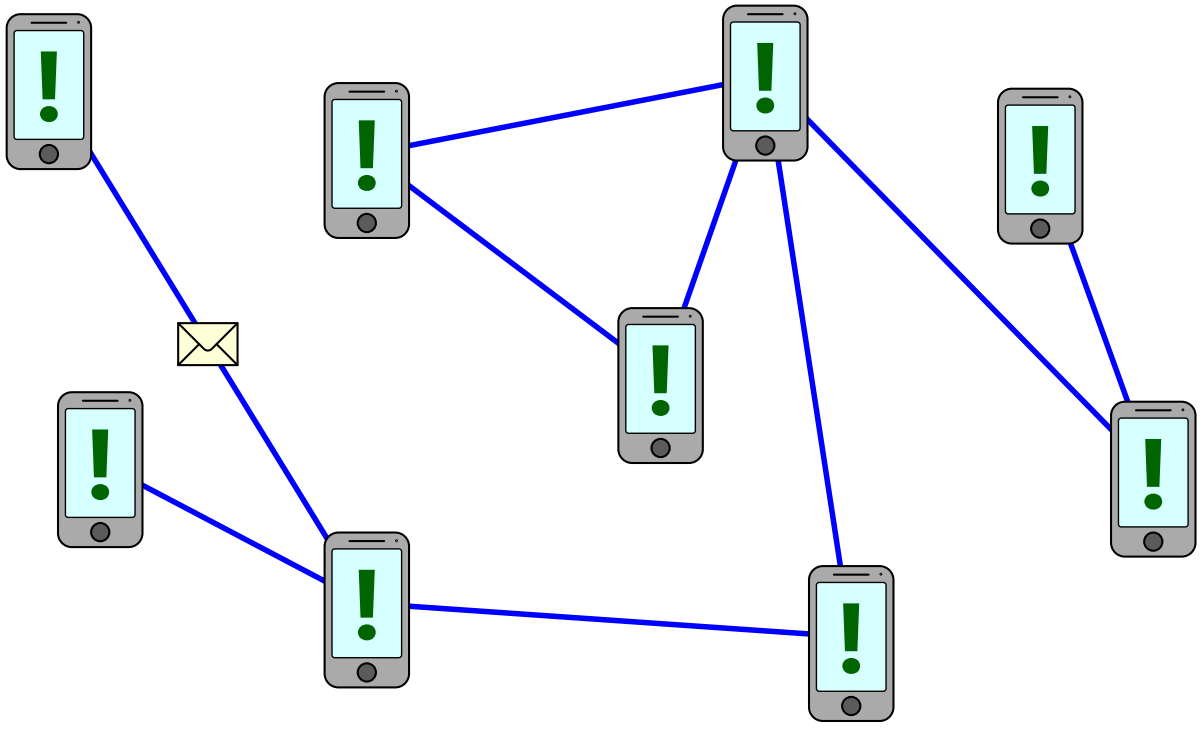
If the network is connected at all rounds, every news reaches every agent in at most $n - 1$ rounds.



Hence, whenever two agents interact, all agents will know it within $n - 1$ rounds (and it will show in their views of the history tree).

Propagation of information

If the network is connected at all rounds, every news reaches every agent in at most $n - 1$ rounds.

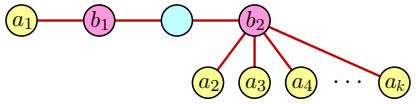


Hence, whenever two agents interact, all agents will know it within $n - 1$ rounds (and it will show in their views of the history tree).

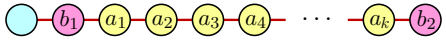
Terminating Algorithm

Counterexample for termination

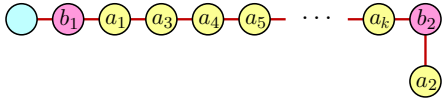
Round 1



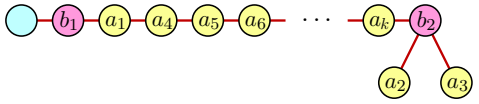
Round 2



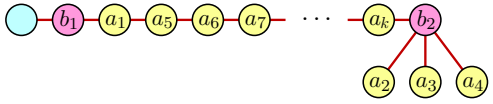
Round 3



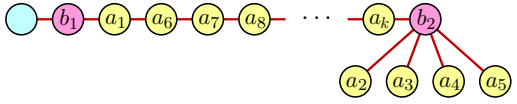
Round 4

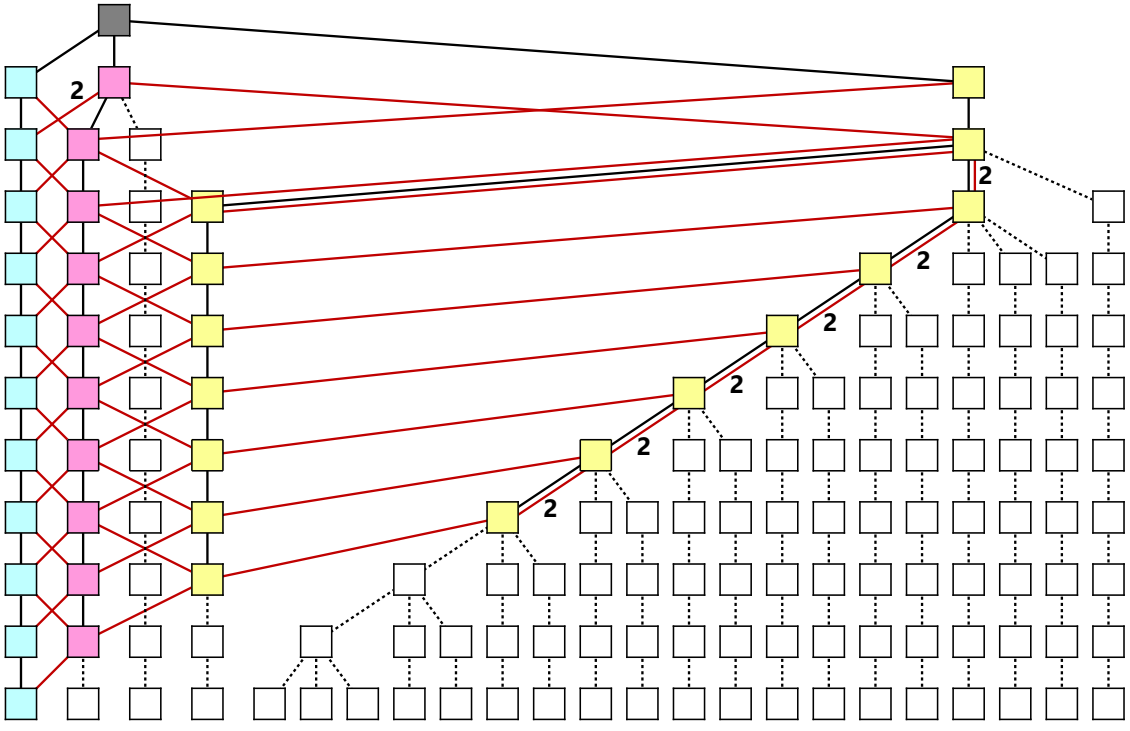


Round 5



Round 6





Terminating algorithm: Overview

We will give a *terminating* algorithm for the Counting Problem.

The algorithm is as follows:

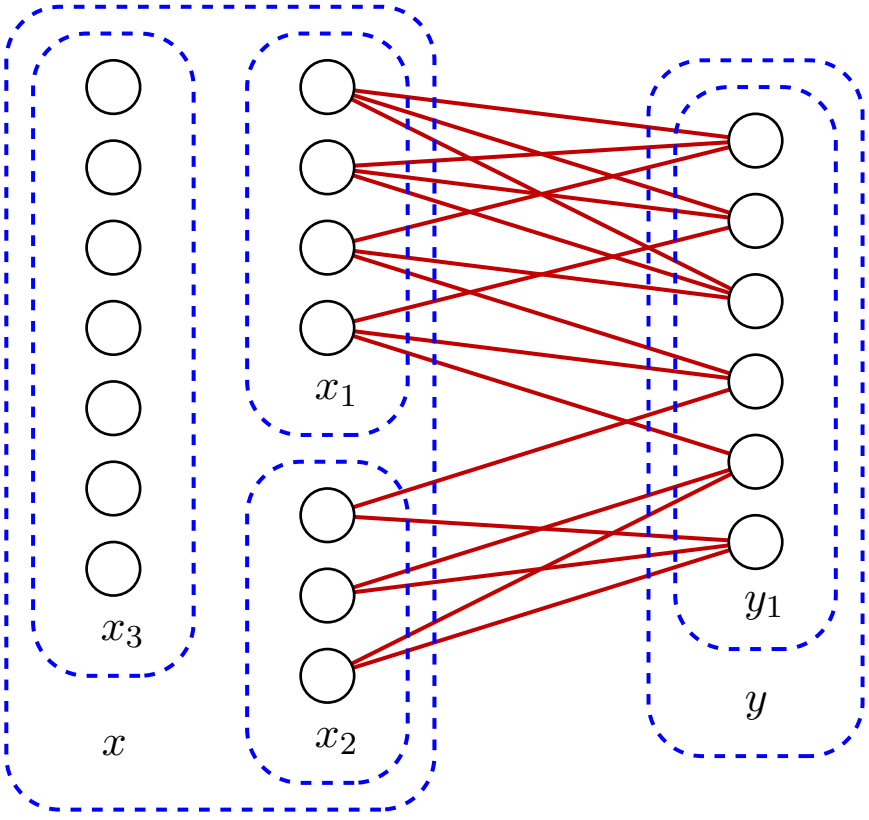
- Use the Leader's observations to make guesses on anonymities.
- In any set of n guesses, we can always identify a correct one.
- Once we have identified $n-1$ correct guesses, we can use some of them to make new guesses on anonymities.
- Repeat until we have the anonymity of all visible branches of the history tree: this gives an estimate n' on n.
- Wait n' rounds to confirm the estimate; if correct, terminate.

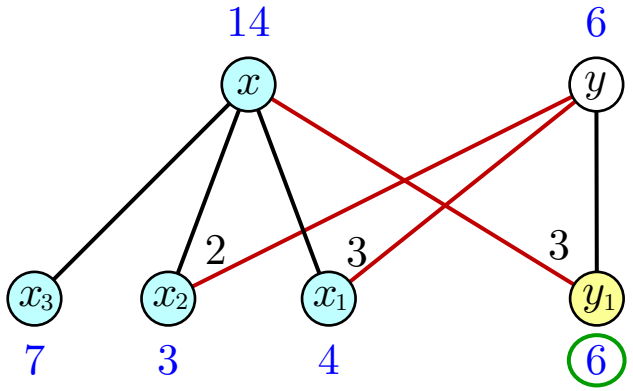
Theorem

The Generalized Counting Problem can be solved in $3n-2$ rounds with explicit termination.

Guessing anonymities

Suppose we know the anonymities of a node x and its children. If some of the agents represented by x have observed agents represented by y, we can *guess* the anonymity of a child of y.



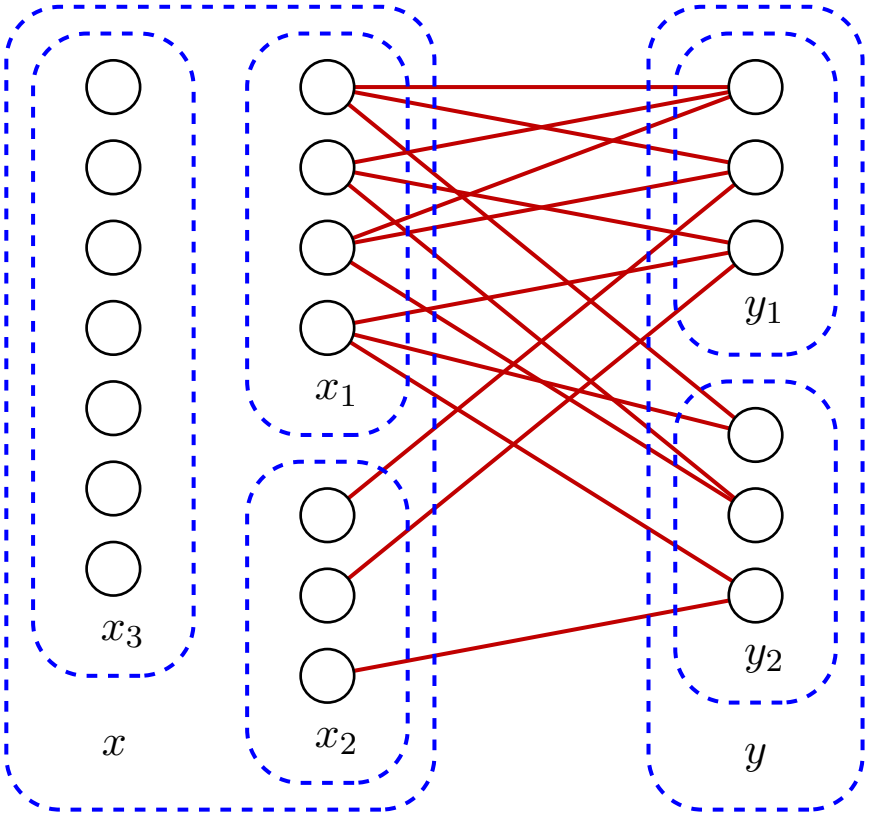


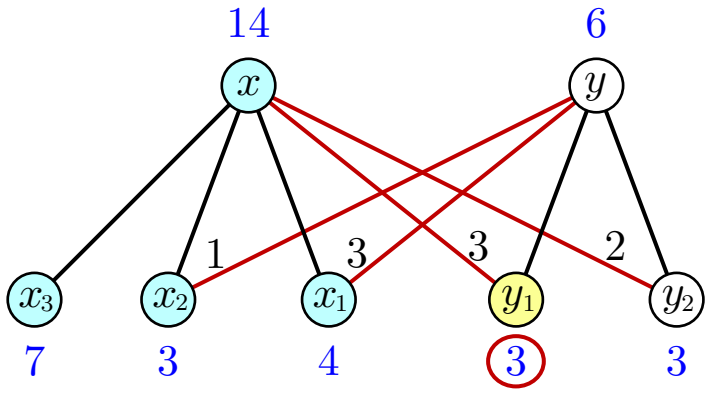
Guess on y_1: $\dfrac{4 \cdot 3 + 3 \cdot 2}{3} = 6$

If only one child of y has seen x, then the guess is *correct*.

Guessing anonymities

Suppose we know the anonymities of a node x and its children. If some of the agents represented by x have observed agents represented by y, we can *guess* the anonymity of a child of y.



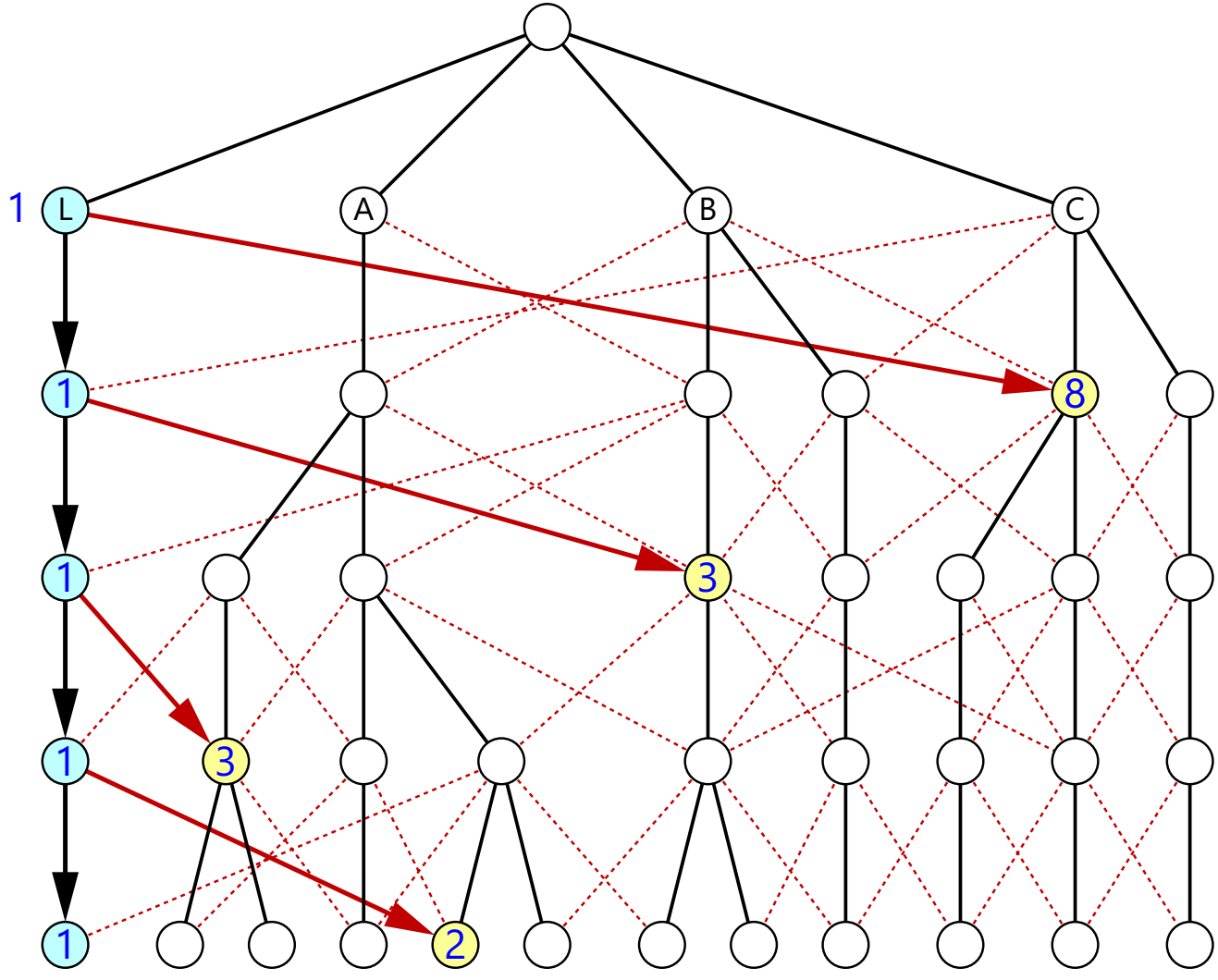


Guess on y_1: $\dfrac{4 \cdot 3 + 3 \cdot 1}{3} = 5$

Otherwise, the guess is an *overestimation* of the anonymity.

Guessing anonymities from the Leader

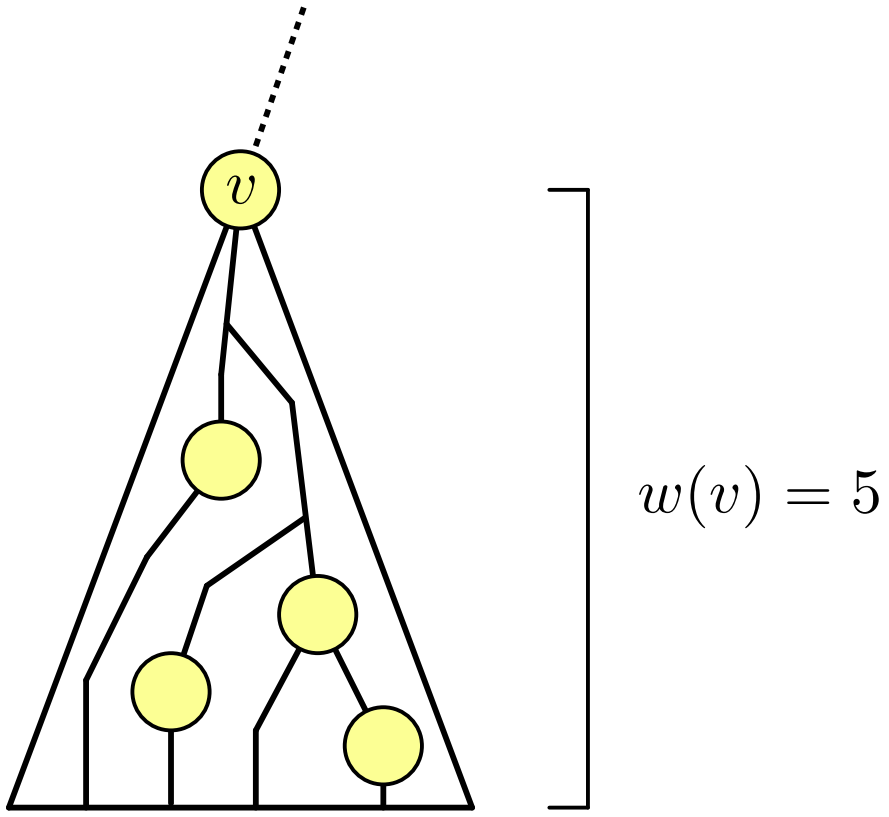
We can make one guess per round using the Leader's observations.



How do we know which guesses are correct?

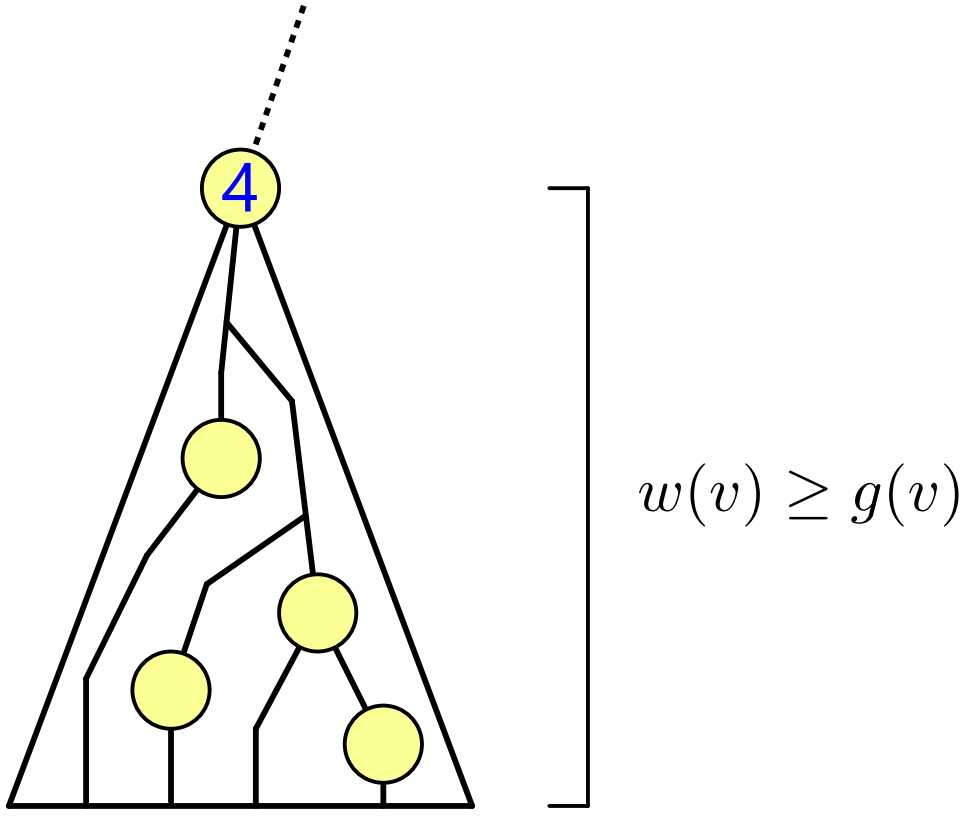
Weight of a node

When a node v has a guess, we define its *weight* $w(v)$ as the number of nodes in the subtree hanging from v that have guesses.



Weight of a node

A node v is *heavy* if its weight $w(v)$ is at least as large as the value of its guess $g(v)$.



Limiting theorem

We denote by $a(v)$ the anonymity of a node v, by $g(v)$ a guess on $a(v)$, and by $w(v)$ the weight of v.

Theorem

If all guesses are on different rounds and $w(v) > a(v)$, then some descendants of v are heavy.

Proof. By well-founded induction on $w(v)$.

Let v_1, v_2, ... be the closest descendants of v that have guesses. Of course, $a(v) \geq \sum_i a(v_i)$.

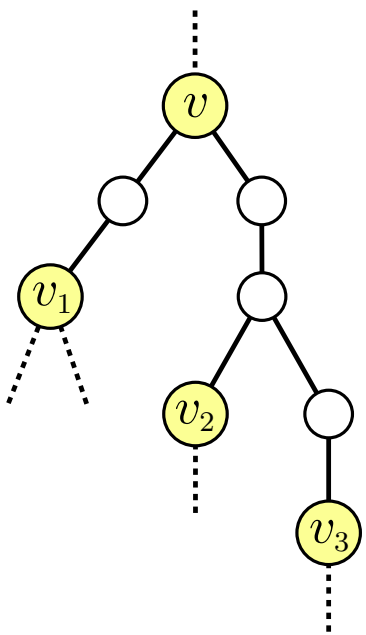
By the inductive hypothesis, $w(v_i) \leq a(v_i)$ for all i.

$w(v)-1 = \sum_i w(v_i) \leq \sum_i a(v_i) \leq a(v) \leq w(v)-1$

Thus, $w(v_i) = a(v_i)$ and $a(v) = \sum_i a(v_i)$.

The deepest node v_d has no siblings, because all guesses are on different rounds.

Hence $g(v_d) = a(v_d) = w(v_d)$, and v_d is heavy. □



Criterion of correctness

Corollary

If v is heavy and no descendant of v is heavy, then $g(v) = a(v)$.

Proof. By assumption, $g(v) \leq w(v)$.

By the limiting theorem, $w(v) \leq a(v)$.

Guesses never underestimate anonymities, and so $a(v) \leq g(v)$.

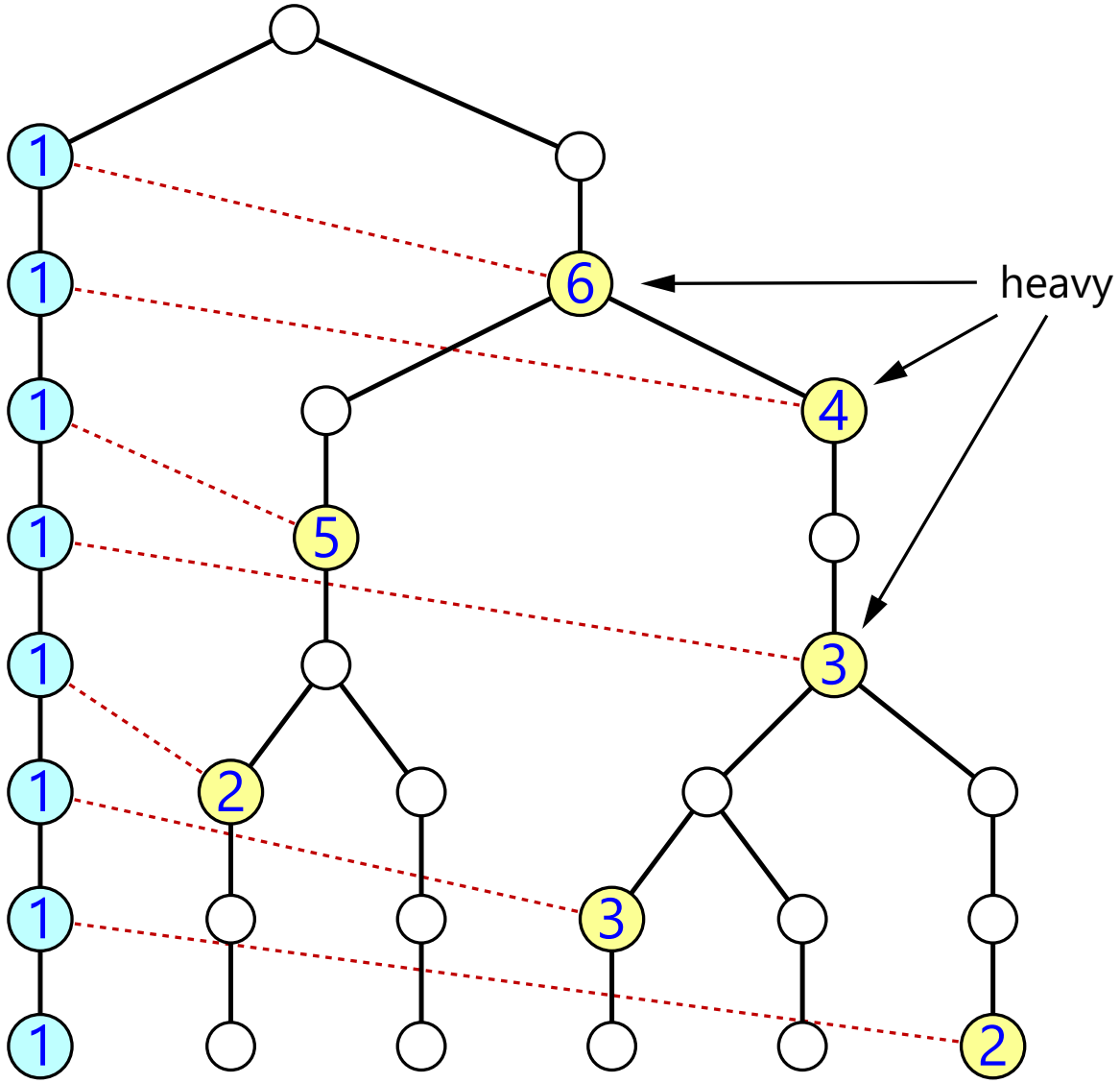
$g(v) \leq w(v) \leq a(v) \leq g(v)$, hence $g(v) = a(v)$. □

This corollary gives agents a criterion to determine when a guess is necessarily correct: **If v is heavy and no descendants of v are heavy, then the guess on v is correct.**

Moreover, by the limiting theorem, such a node v is found by the time there are n guesses in total.

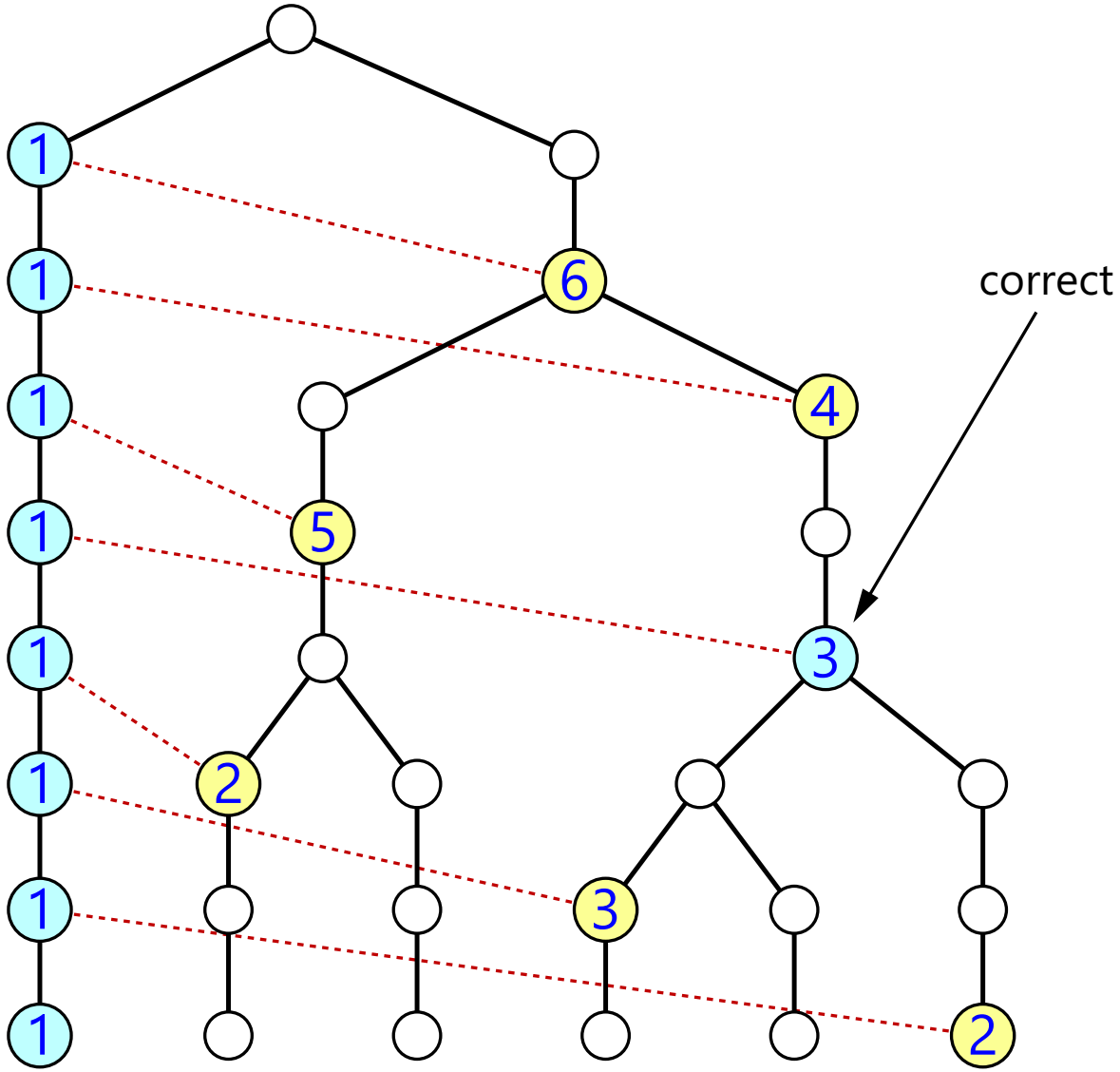
Criterion of correctness: Example

Any agent with this view is able to determine which guess is necessarily correct:



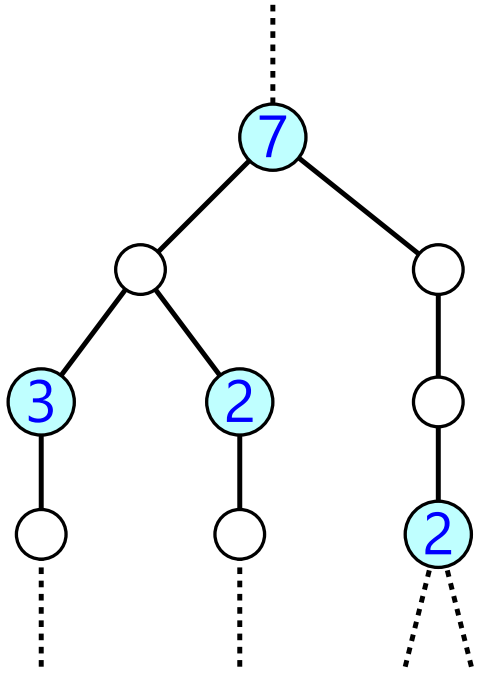
Criterion of correctness: Example

Any agent with this view is able to determine which guess is necessarily correct:



Propagation of guesses

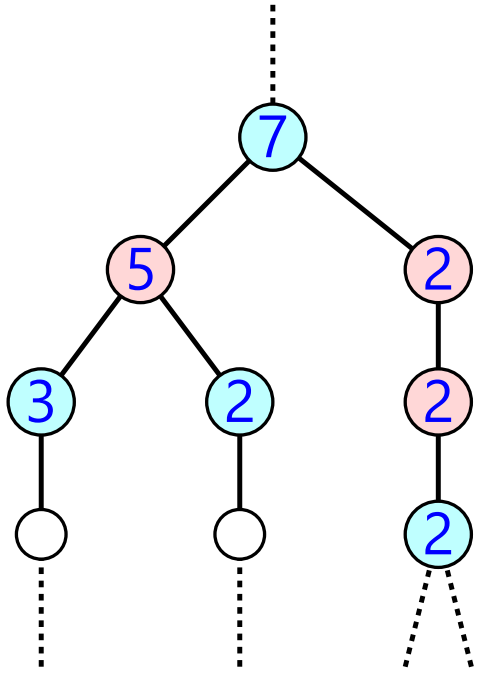
An *island* is a connected component of (a view of) the history tree that contains no leaves and does not contain the root.



Suppose that the nodes with necessarily correct guesses bound an island in the history tree.

Propagation of guesses

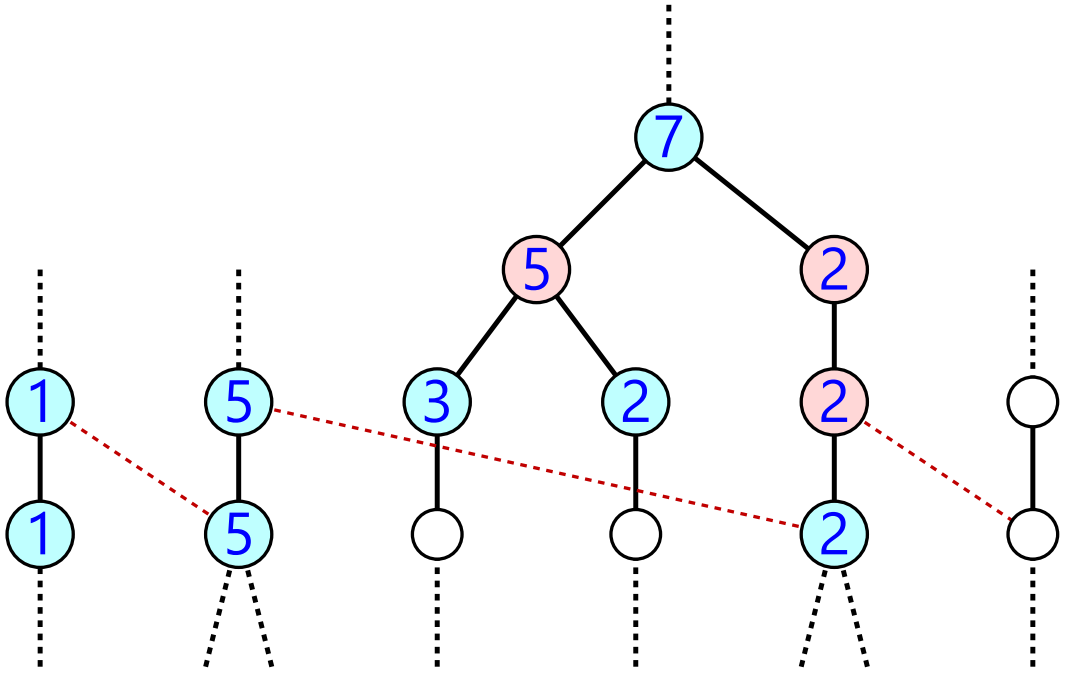
An *island* is a connected component of (a view of) the history tree that contains no leaves and does not contain the root.



If the anonymity of the top node is the sum of the bottom ones, then we can infer the anonymities of all the nodes in the island.

Propagation of guesses

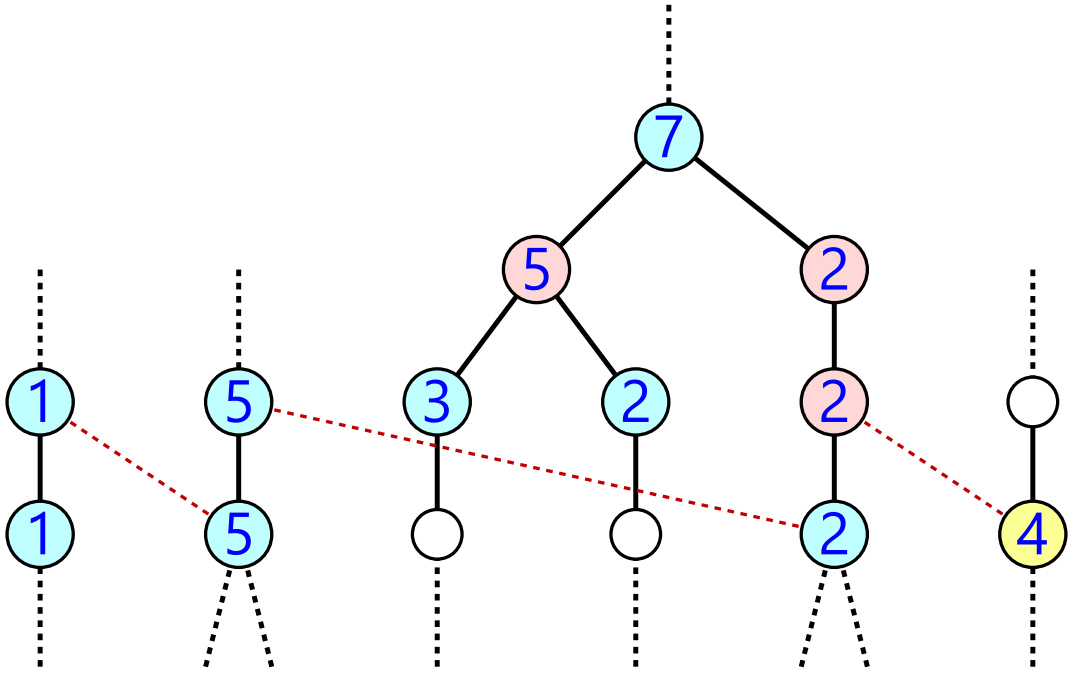
An *island* is a connected component of (a view of) the history tree that contains no leaves and does not contain the root.



Since the network is connected at every round, we can make a new guess from one of the nodes in the island.

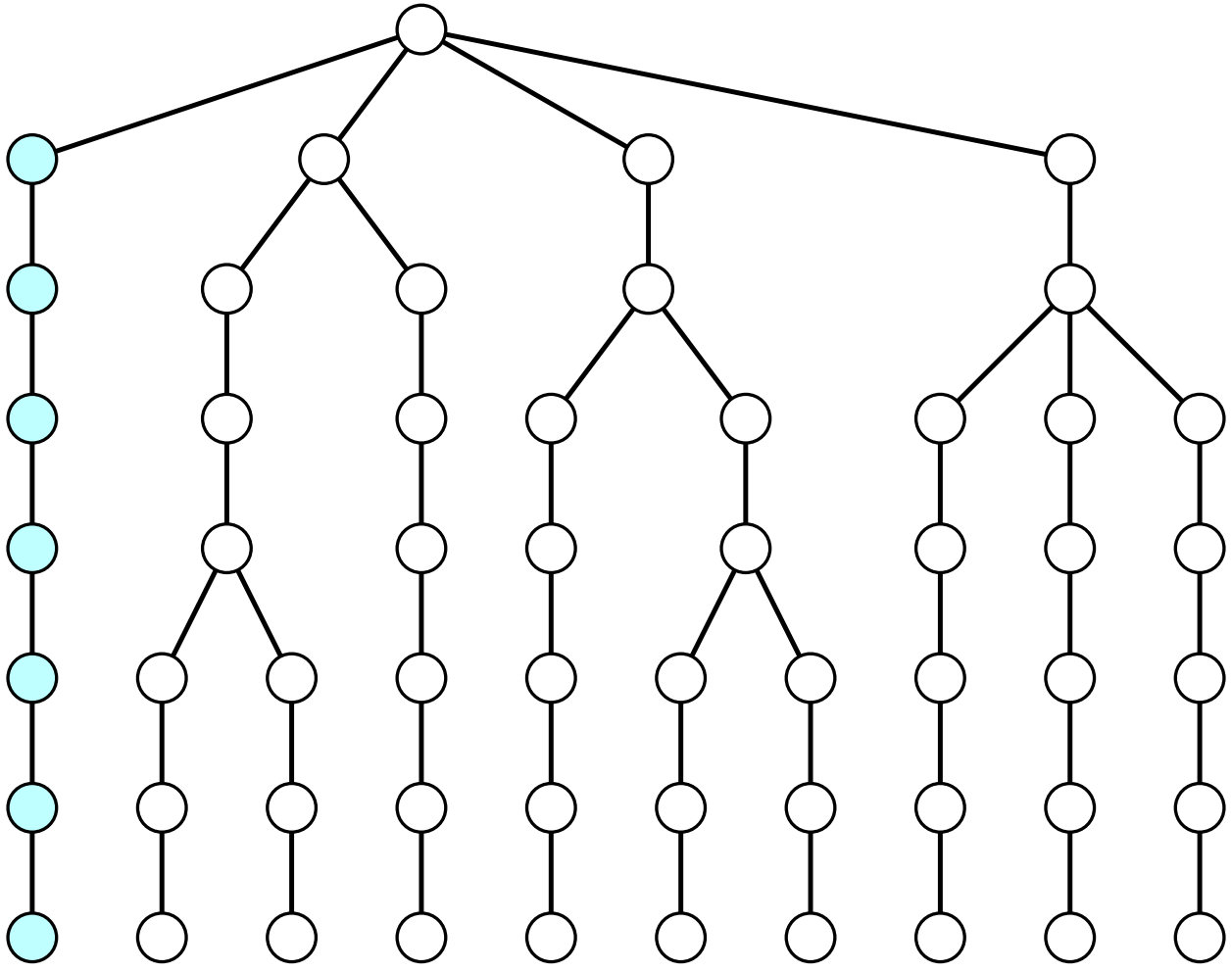
Propagation of guesses

An *island* is a connected component of (a view of) the history tree that contains no leaves and does not contain the root.



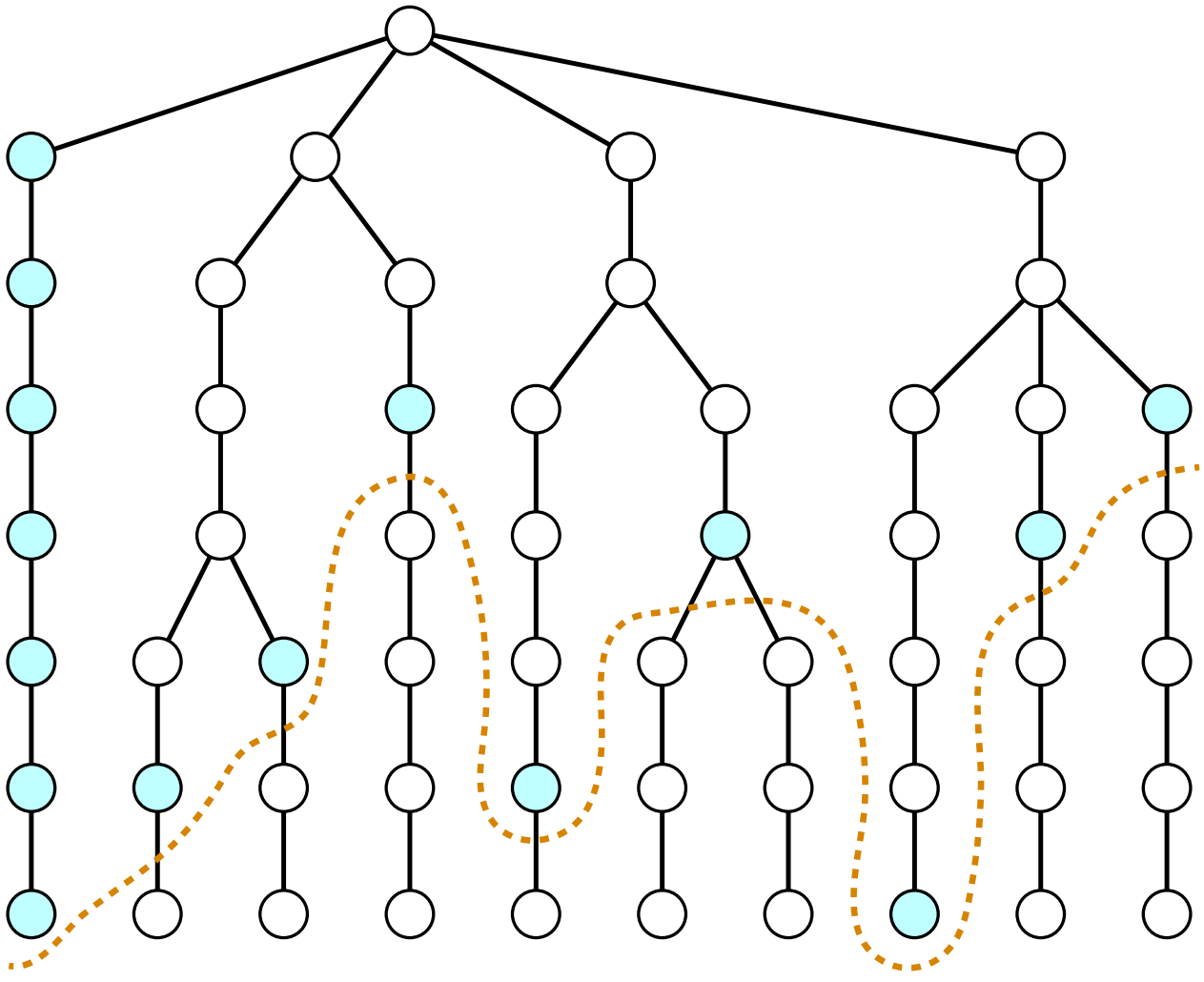
Since the network is connected at every round, we can make a new guess from one of the nodes in the island.

Propagation of guesses



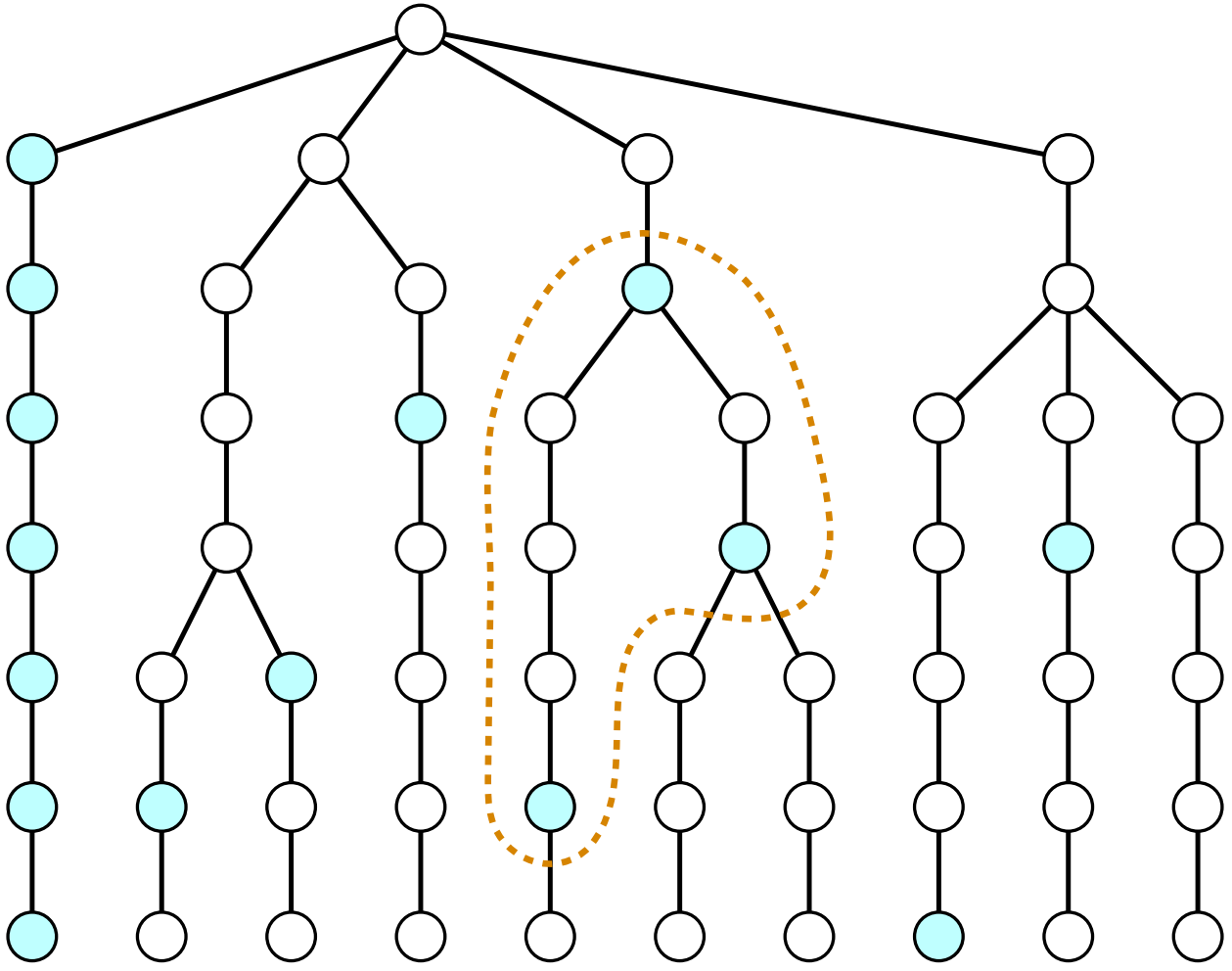
Suppose that there are $n-1$ nodes with necessarily correct guesses (other than the Leader ones). There are two cases:

Propagation of guesses



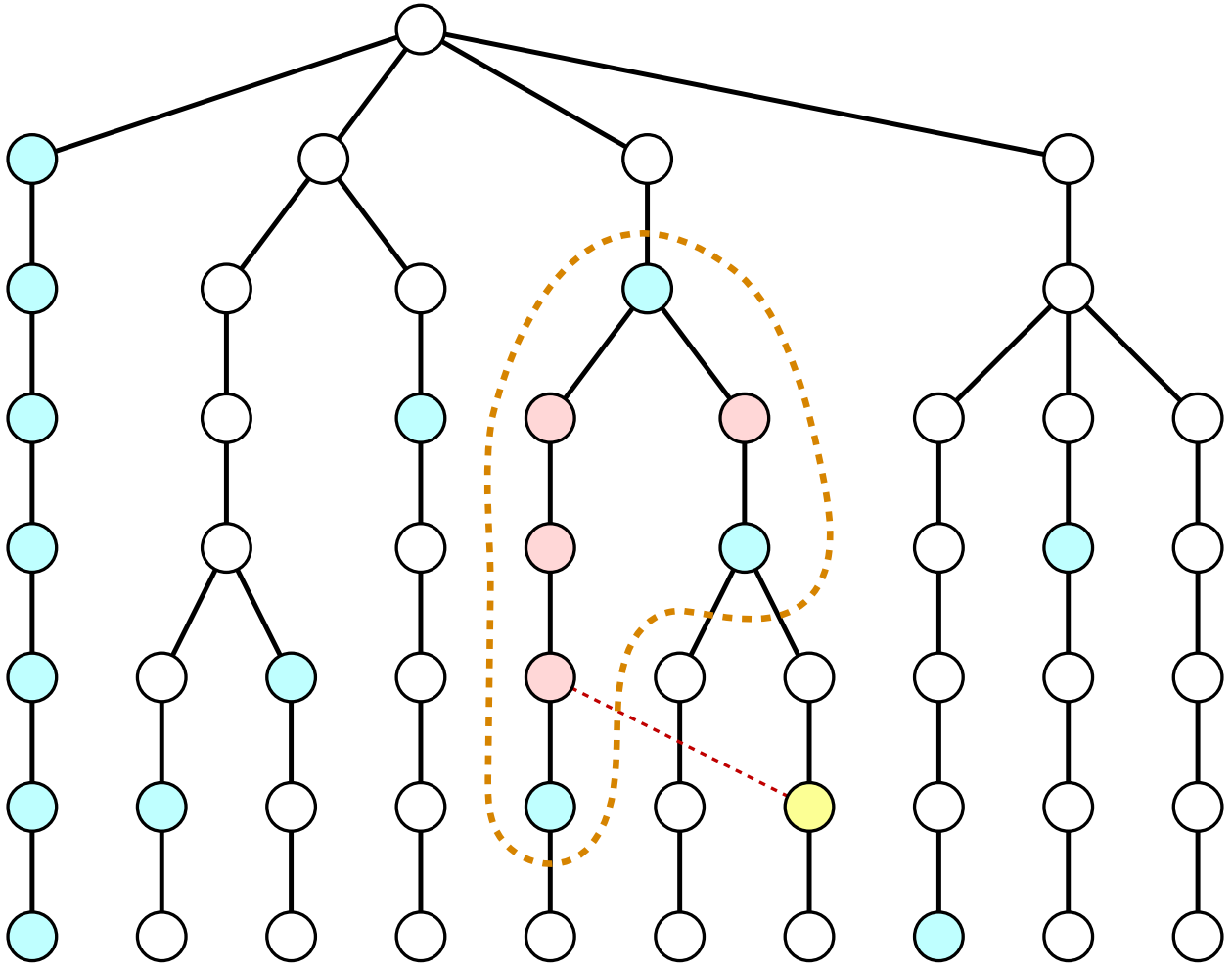
Either these nodes determine a *cut* of the history tree, in which case we have an estimate n' on n, given by their sum...

Propagation of guesses



...Or else, some of these nodes determine an island, which allows us to make a new guess, and so on.

Propagation of guesses



...Or else, some of these nodes determine an island, which allows us to make a new guess, and so on.

Dynamics of new guesses

Summarizing, we have two "buffers":

- A buffer of $n-1$ overestimating guesses (yellow nodes),
- A buffer of $n-2$ necessarily correct guesses (blue nodes).

Once the first buffer is full, every new guess (yellow node) allows us to determine a correct guess (blue node), by the limiting theorem.

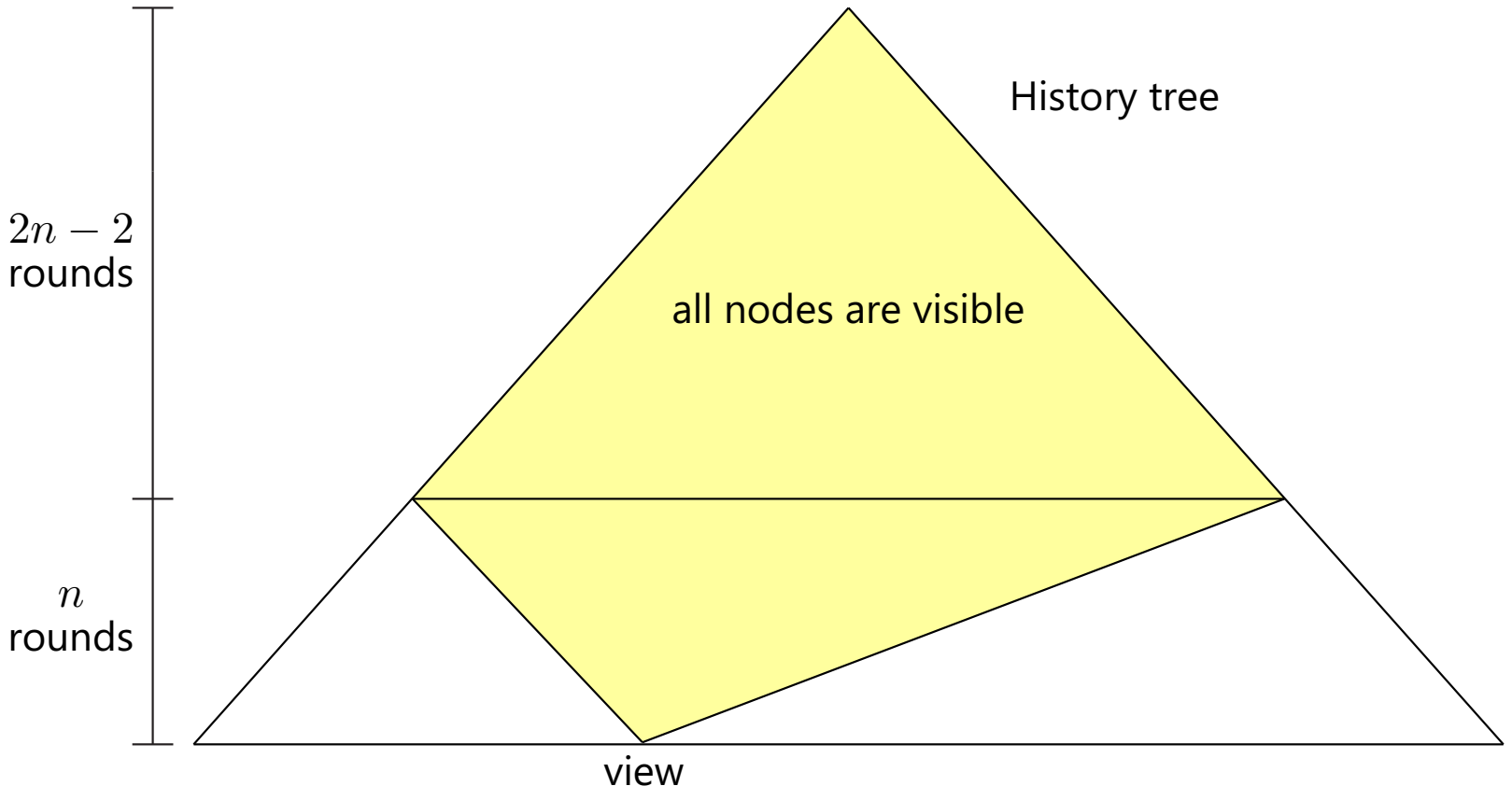
Once the second buffer is full, every new correct guess (blue node) either creates a new island or splits an island in two; in both cases, we can make a new guess (yellow node).

Therefore, within $2n-2$ rounds, the chain of guesses "snowballs" and generates enough guesses to determine a *cut* of the history tree, which in turn yields an *estimate* $n' \leq n$.

Termination condition

Once we have a cut and an estimate $n' \leq n$, we wait n' rounds.

If $n' < n$, a new node appears in the first levels of the history tree.



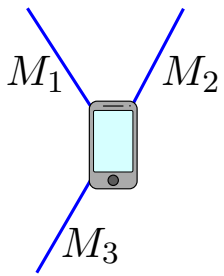
If $n' = n$, then no new nodes appear, and the algorithm terminates.

Disconnected Networks

T-interval-disconnected networks

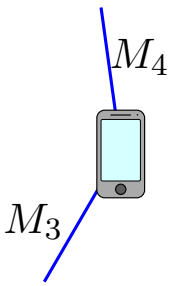
Any algorithm for $T = 1$ can be adapted to networks with $T > 1$, assuming T is known by all agents.

Round $kT + 1$



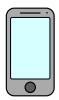
Message list
$M_1 \times 1$
$M_2 \times 1$
$M_3 \times 1$

Round $kT + 2$



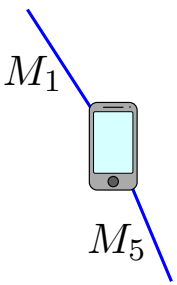
Message list
$M_1 \times 1$
$M_2 \times 1$
$M_3 \times 2$
$M_4 \times 1$

Round $kT + 3$



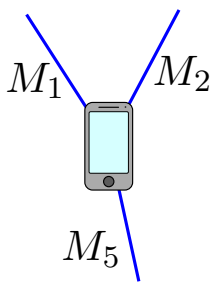
Message list
$M_1 \times 1$
$M_2 \times 1$
$M_3 \times 2$
$M_4 \times 1$

Round $kT + 4$



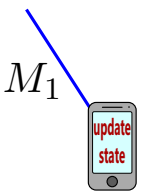
Message list
$M_1 \times 2$
$M_2 \times 1$
$M_3 \times 2$
$M_4 \times 1$
$M_5 \times 1$

Round $kT + 5$



Message list
$M_1 \times 3$
$M_2 \times 2$
$M_3 \times 2$
$M_4 \times 1$
$M_5 \times 2$

Round $kT + T$

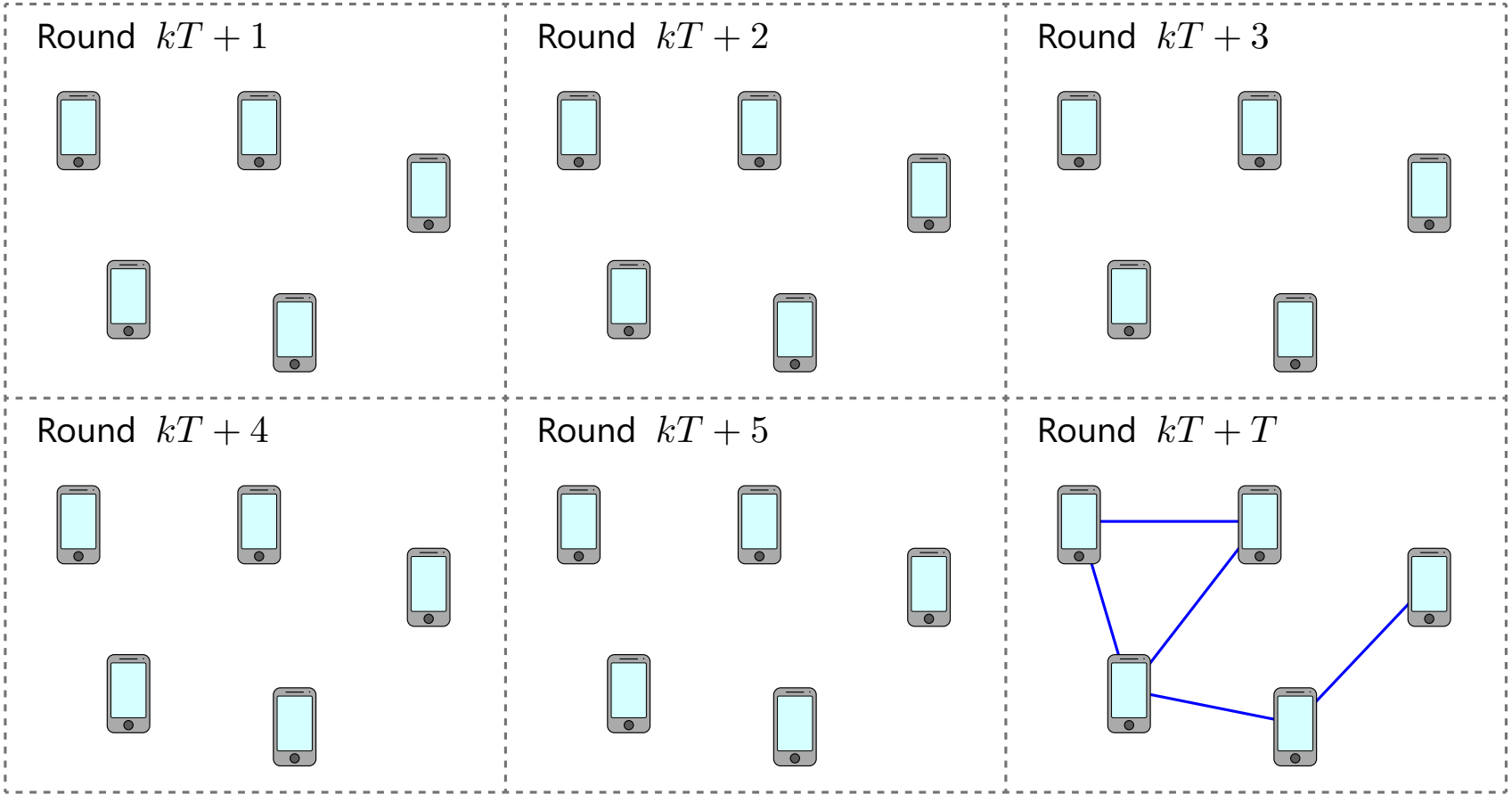


Message list
$M_1 \times 4$
$M_2 \times 2$
$M_3 \times 2$
$M_4 \times 1$
$M_5 \times 2$

Each agent accumulates messages for T rounds, and then updates its state all at once. Hence, the running time is multiplied by T.

T-interval-disconnected networks

This is the best we can do: Consider, for instance, a network that contains no links for $T-1$ out of every T rounds.



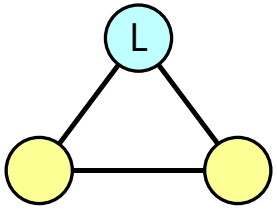
Thus, the Counting Problem has a lower bound of $2Tn$ rounds.

Multiple Leaders

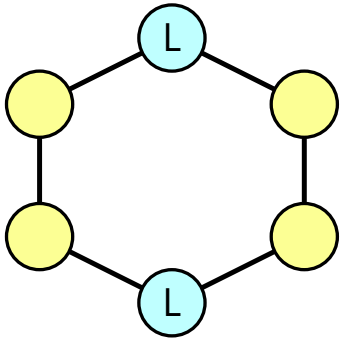
Multiple Leaders

The Counting Problem is unsolvable with no knowledge on the number of Leaders, ℓ.

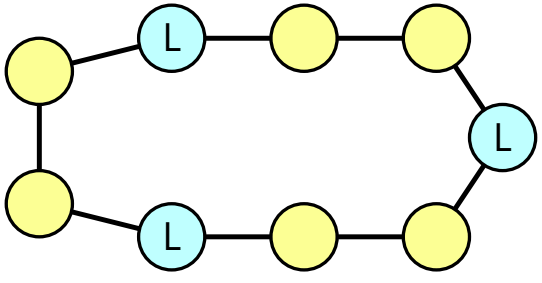
System 1



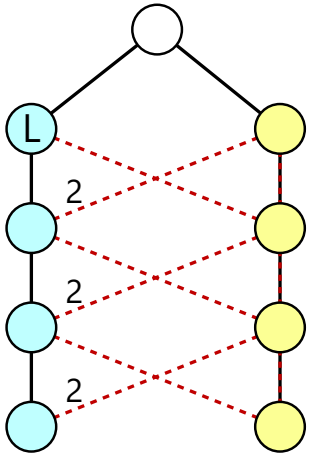
System 2



System 3



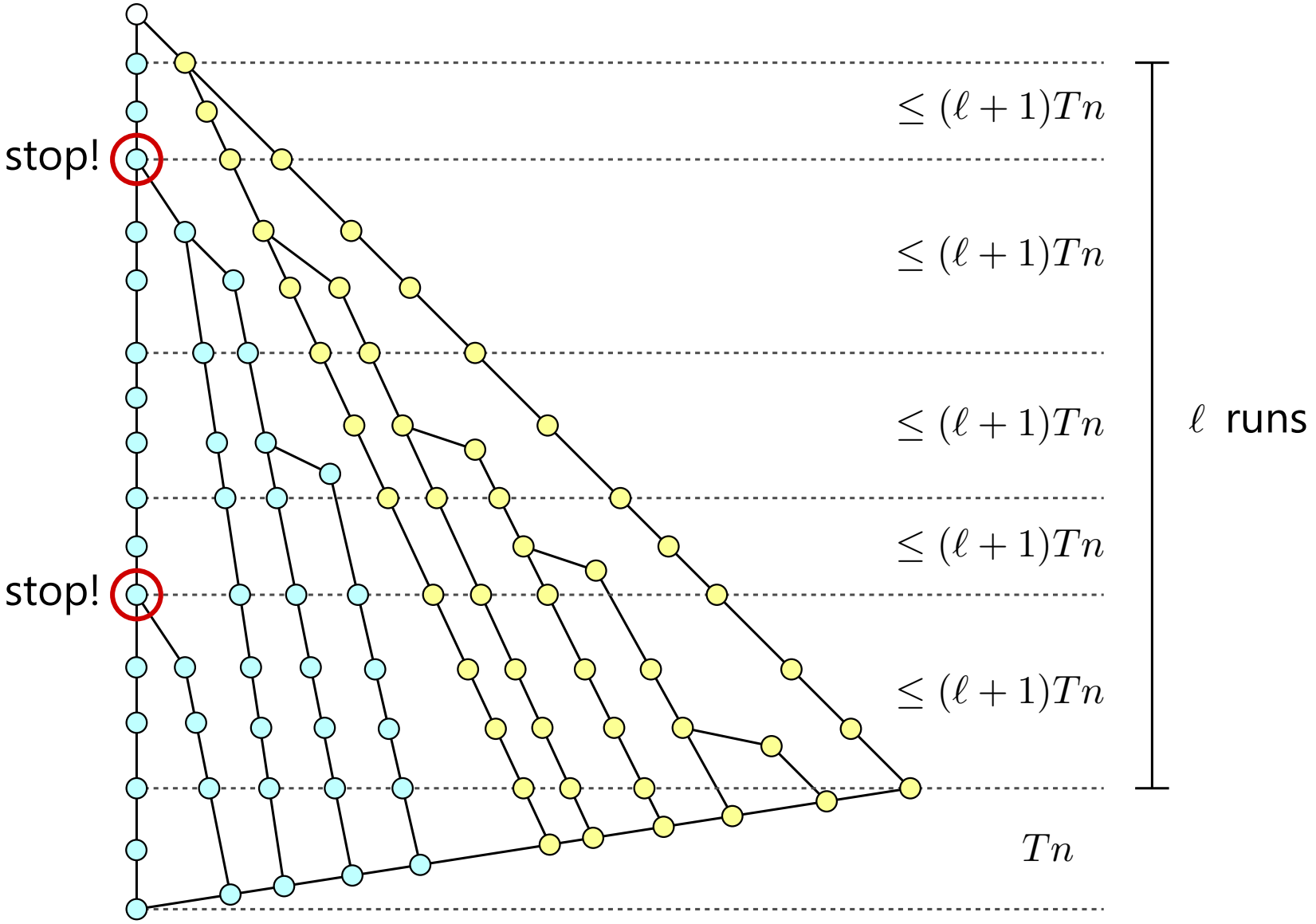
History tree



In the above (static) networks, the history tree is the same.

Multiple Leaders

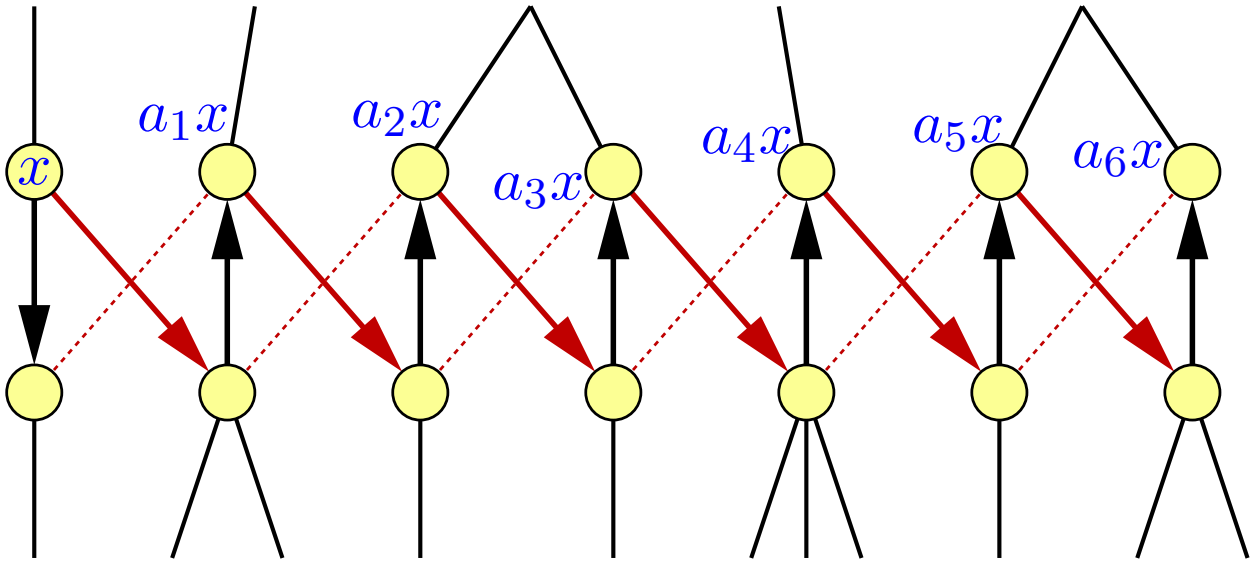
With $\ell > 1$ of Leaders, we do ℓ *runs* similar to the $\ell = 1$ algorithm.



Leaderless Computation

Leaderless computation

If there is no Leader, we can still use our stabilizing technique and find a level where no nodes branch within the first $n-1$ levels.



However, since no anonymities are given, we can only compute them up to a common factor x. We just assign anonymity x to an arbitrary node, and then express all other anonymities as linear functions of x.

Leaderless computation

Thus, if there is no Leader, we can solve in $2Tn$ rounds all the multi-aggregate functions f such that:

$f(x_i, \{x_1 \times n_1, x_2 \times n_2, \ldots, x_m \times n_m\}) =$
$f(x_i, \{x_1 \times kn_1, x_2 \times kn_2, \ldots, x_m \times kn_m\})$.

We call them *scale-invariant multi-aggregate* functions.

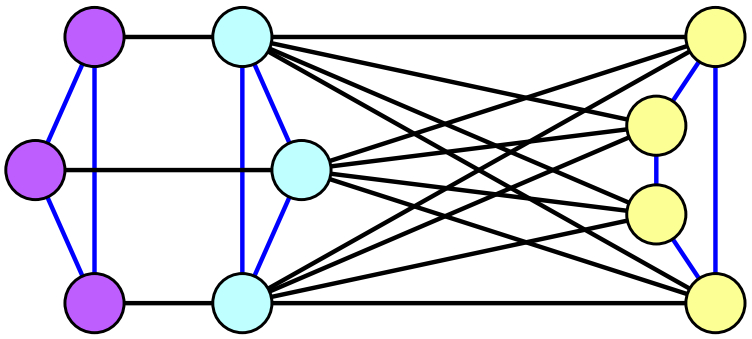
Examples include the mean (cf. *Average Consensus Problem*), variance, median, maximum, mode, and other statistical functions.

In Leaderless networks, we can compute functions that depend only on the "concentration" of each input.

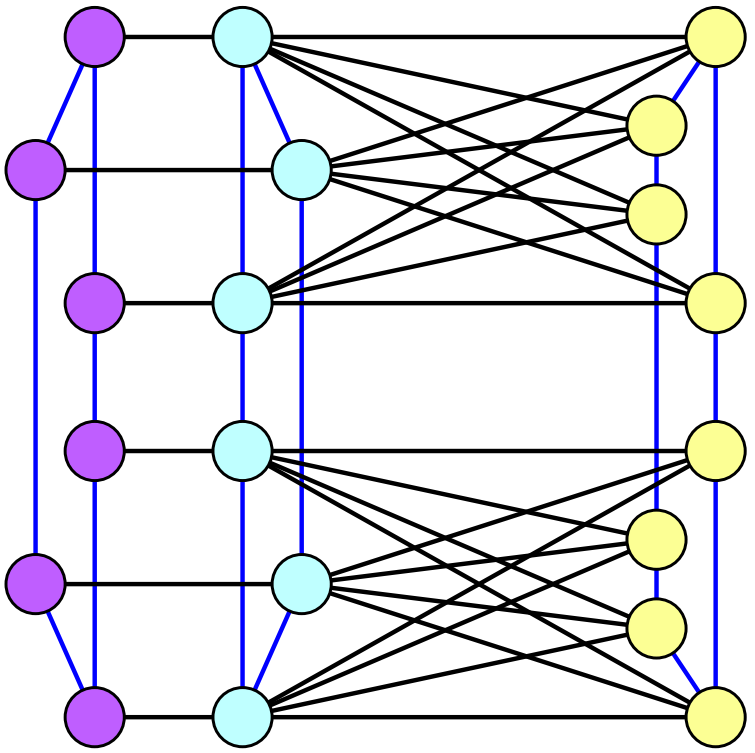
Leaderless computation

The following example shows that no other function can be computed without a Leader: We can multiply all anonymities by any integer factor ≥ 2 and get the same history tree.

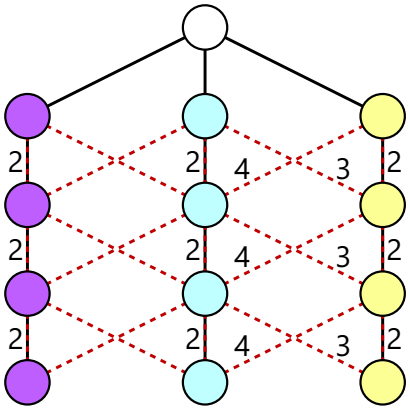
System 1



System 2

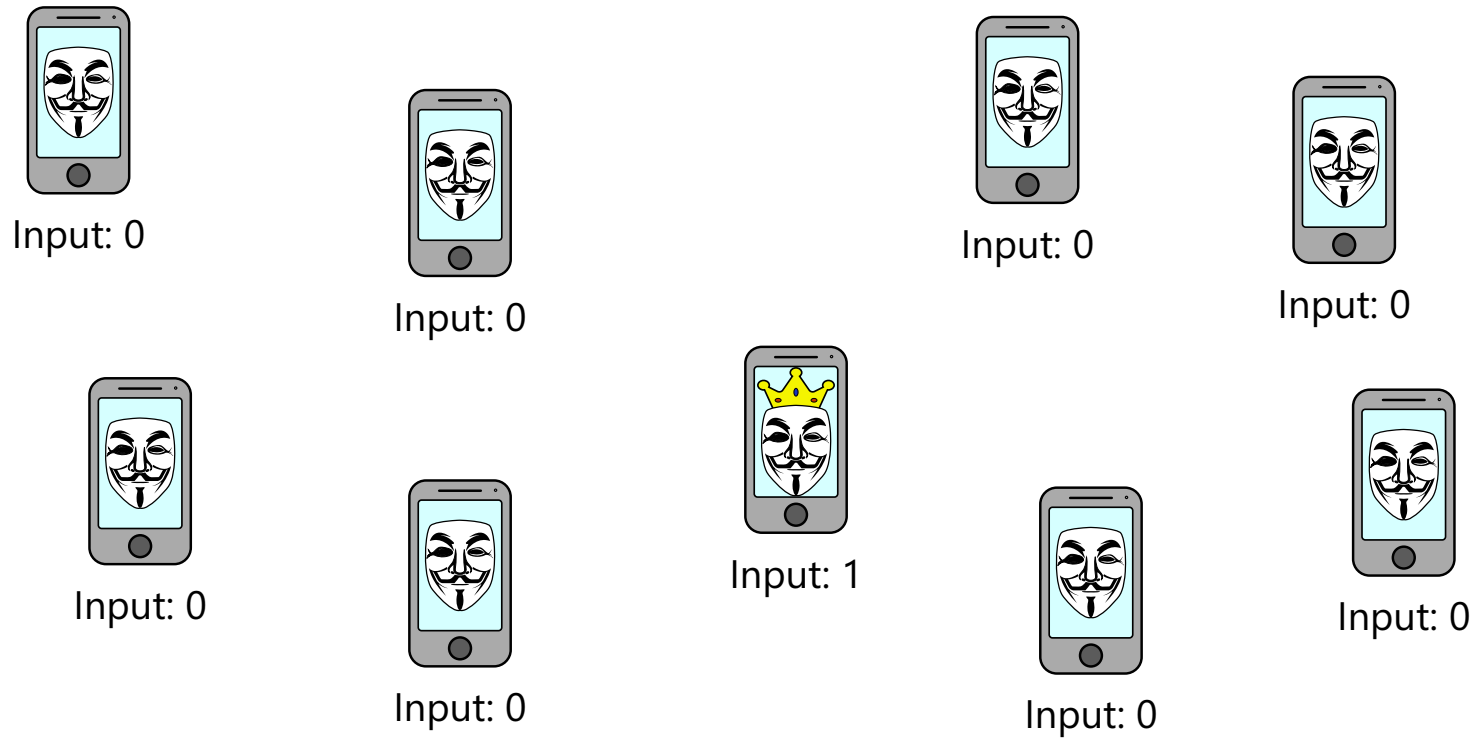


History tree



Leaderless computation

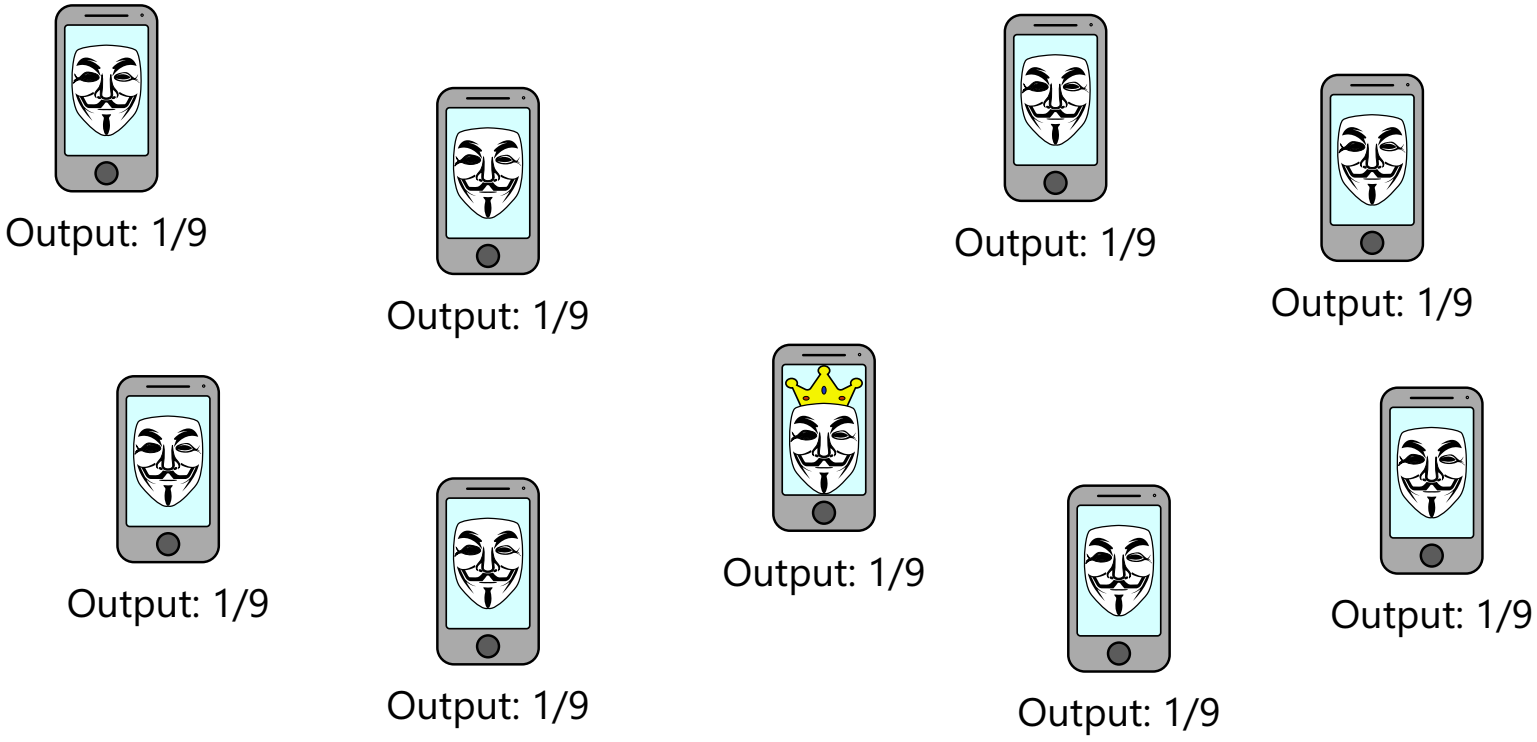
Also, $2Tn$ is a lower bound on the Average Consensus Problem.



Indeed, if we assign input 1 to one agent and 0 to all other agents, the Average Consensus Problem becomes equivalent to the Counting Problem with a single Leader.

Leaderless computation

Also, $2Tn$ is a lower bound on the Average Consensus Problem.



Indeed, if we assign input 1 to one agent and 0 to all other agents, the Average Consensus Problem becomes equivalent to the Counting Problem with a single Leader.

Further results

Theorem (Di Luna–V., arXiv 2022)

Any problem that is solvable in a T-interval-disconnected anonymous dynamic network with no Leader has a solution:

- *in $2Tn$ rounds without explicit termination,*
- *in $T(n+N)$ rounds with termination if $N \geq n$ is known.*

$2Tn$ rounds is a lower bound for the Average Consensus Problem.

Theorem (Di Luna–V., arXiv 2022)

Any problem that is solvable in a T-interval-disconnected anonymous dynamic network with a known number $\ell \geq 1$ of Leaders has a solution:

- *in $2Tn$ rounds without explicit termination,*
- *in $(\ell^2 + \ell + 1)Tn$ rounds with termination.*

$2Tn$ rounds is a lower bound for the Counting Problem.

Future work: What if message size is limited to $O(\log n)$?