# Seminar 2 – Anonymous Networks: History Trees and Applications

## Distributed Computing in Anonymous Dynamic Systems

Giovanni Viglietta

Rome – March 7, 2024

**Syllabus**

- Anonymous Networks
  - Introduction and basic algorithms for static networks
  - Dynamicity and history trees
  - Optimal computation in networks with and without leaders
  - Computation in dynamic congested networks
- Population Protocols
  - Introduction and basic algorithmic techniques
  - Leader election in Mediated Population Protocols
- Mobile Robots
  - Gathering and Pattern Formation in the plane
  - Meeting in a polygon by oblivious robots

**Exam**

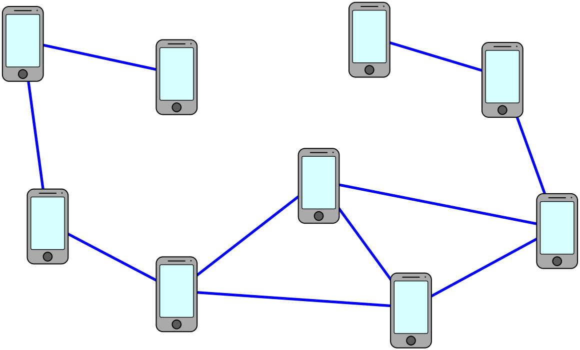
Pre-recorded 10-minute presentation video on one of the papers that will be suggested at the end of the course.

# Today's seminar

- Introduction to history trees
- Counting in dynamic networks
- Average consensus in dynamic networks
- Leader election with knowledge of $n$

# Dynamic networks

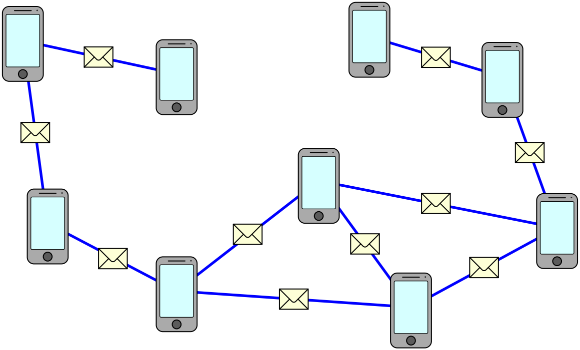
In a *dynamic network*, some machines (or agents) are connected with each other through links that may change over time.



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

# Dynamic networks

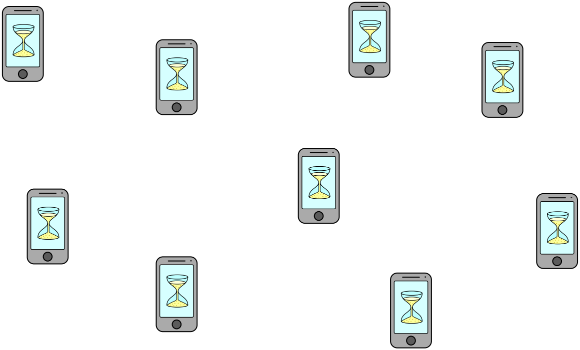
In a *dynamic network*, some machines (or agents) are connected with each other through links that may change over time.



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

# Dynamic networks

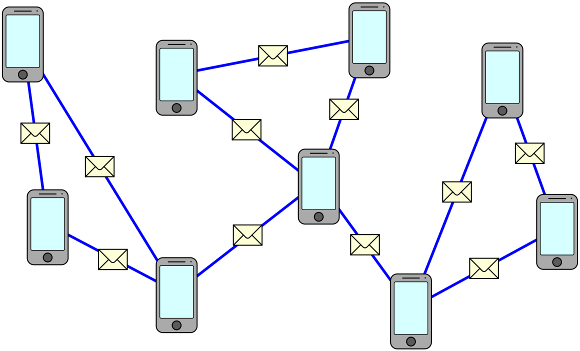
In a *dynamic network*, some machines (or agents) are connected with each other through links that may change over time.



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

# Dynamic networks

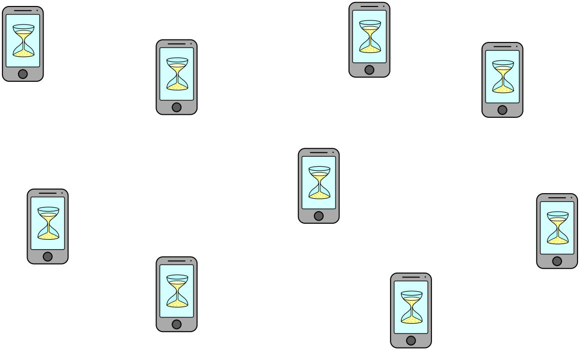
In a *dynamic network*, some machines (or agents) are connected with each other through links that may change over time.



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

# Dynamic networks

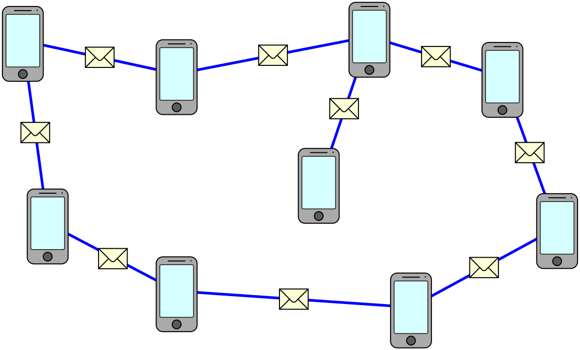
In a *dynamic network*, some machines (or agents) are connected with each other through links that may change over time.



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?

# History Trees

# History trees

A *history tree* is a combinatorial structure that characterizes classes of *indistinguishable agents* and the way they interact in time.

Essentially, two agents $p$ and $q$ are *distinguishable* at round $t$ if:

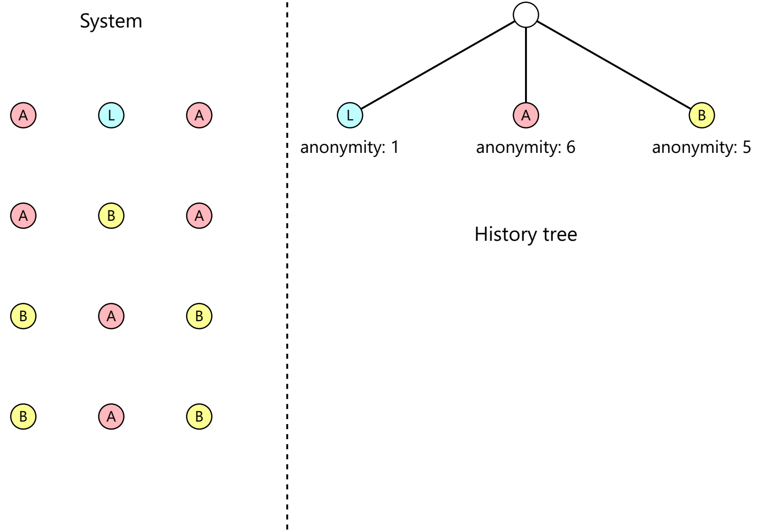
- $t = 0$ and $p$ and $q$ have different inputs, OR
- $t > 0$ and
  - $p$ and $q$ were distinguishable at round $t - 1$, OR
  - $p$ and $q$ receive different multisets of messages at round $t$.

A history tree has a *level* for every round, and each level contains a *node* for every class of indistinguishable agents.

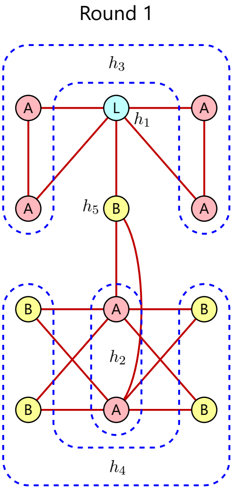
Rephrasing problems in terms of history trees is often a useful algorithmic idea.

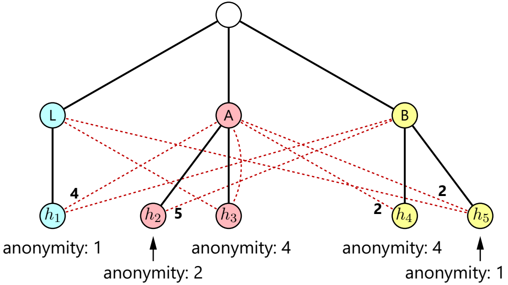
This shifts the problem from anonymous networks to history trees (and their construction).

# Constructing a history tree

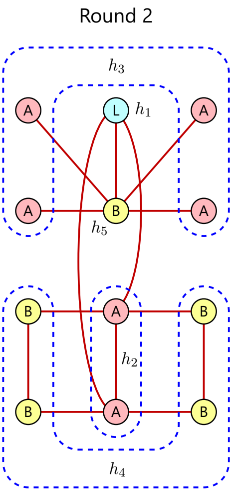


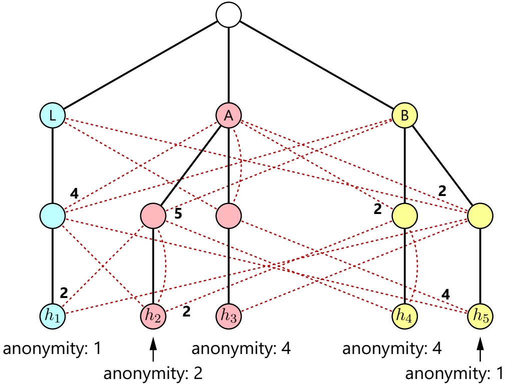
# Constructing a history tree



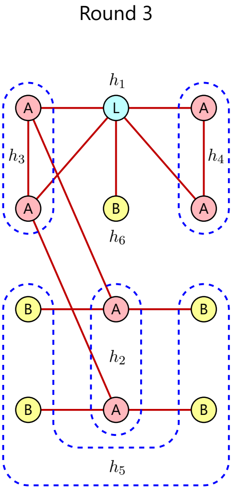



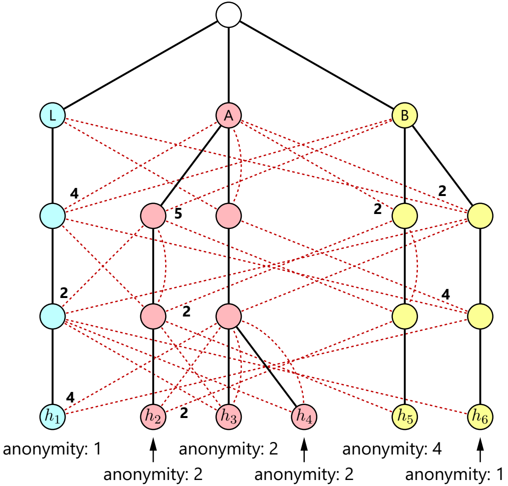
# Constructing a history tree



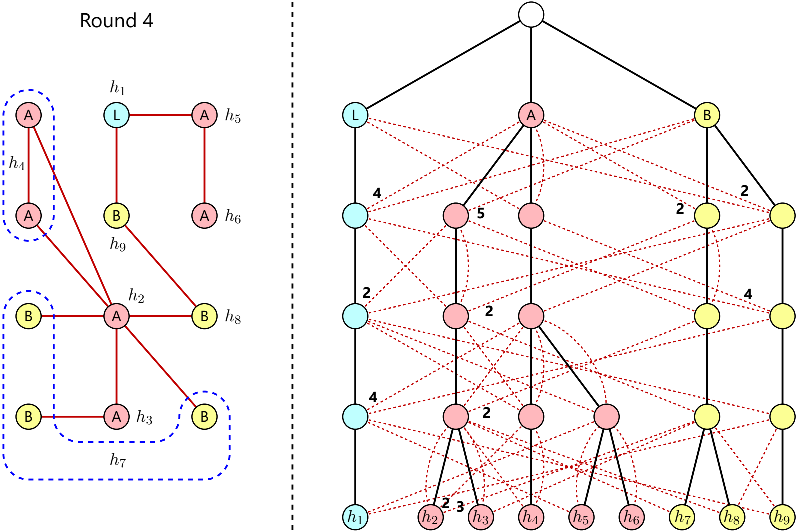



# Constructing a history tree



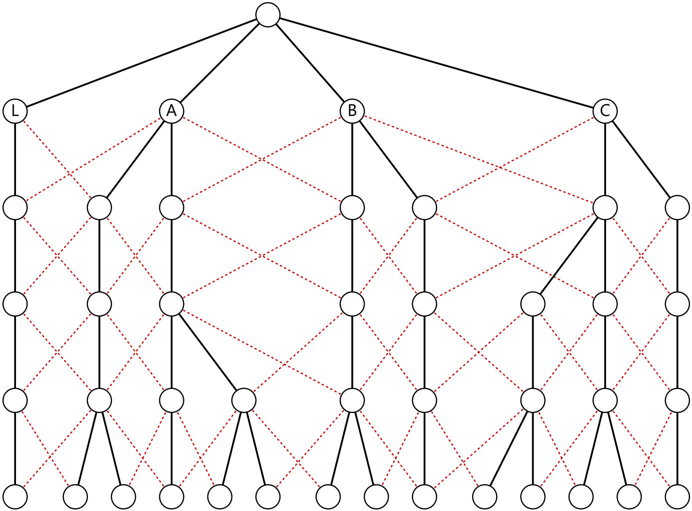



# Constructing a history tree



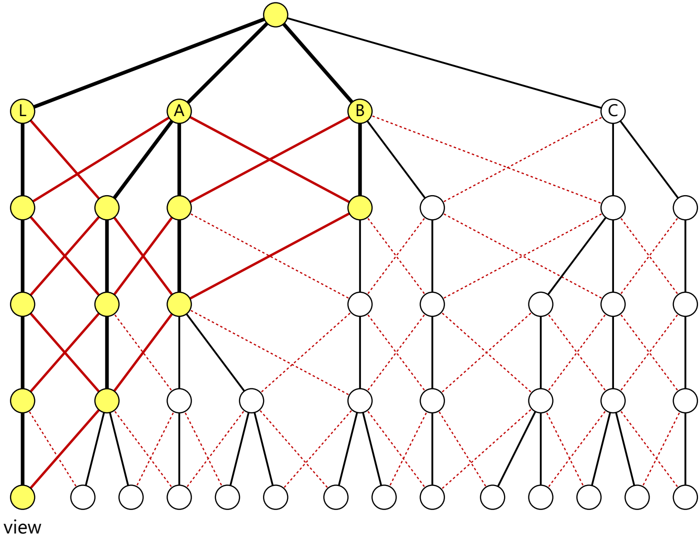
# View of a history tree

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



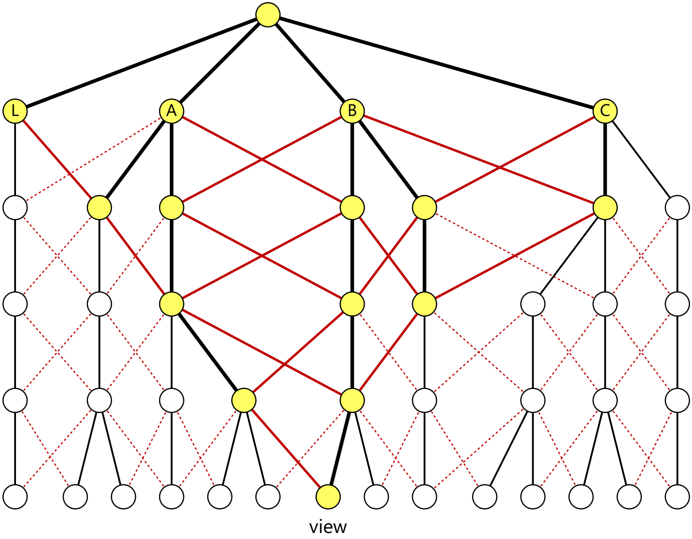
# View of a history tree

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



# View of a history tree

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



# Views as internal states and messages

An agent's view summarizes its whole *history* up to some round. It contains all the information the agent has learned about the network up to that point.

**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.*

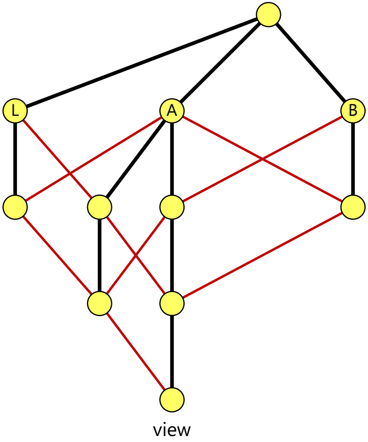
At round $t$, the size of a view is only $O(tn^2 \log n)$ bits.

**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.*

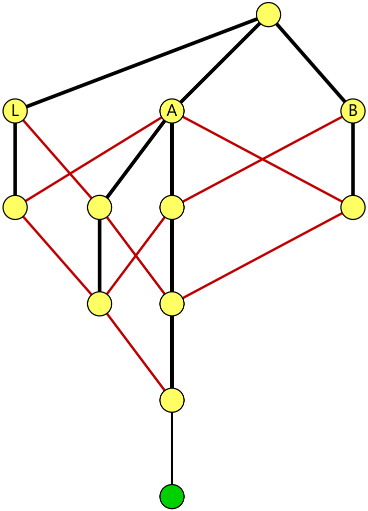
# Updating the view

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



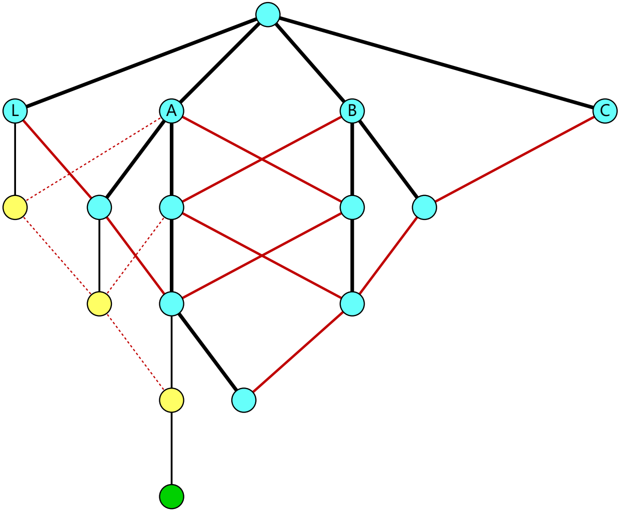
# Updating the view

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



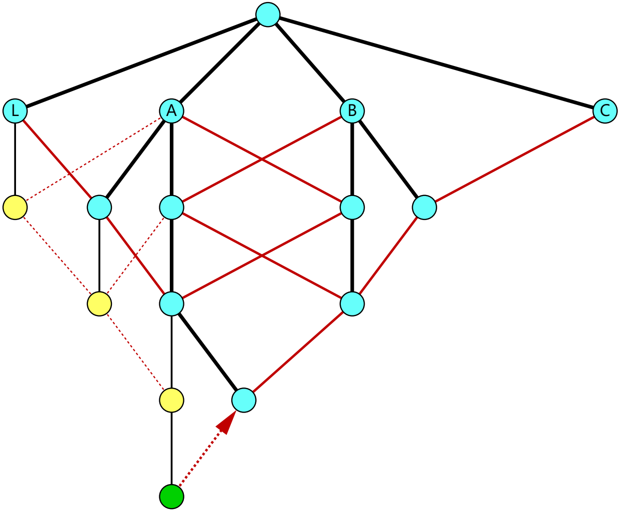
# Updating the view

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



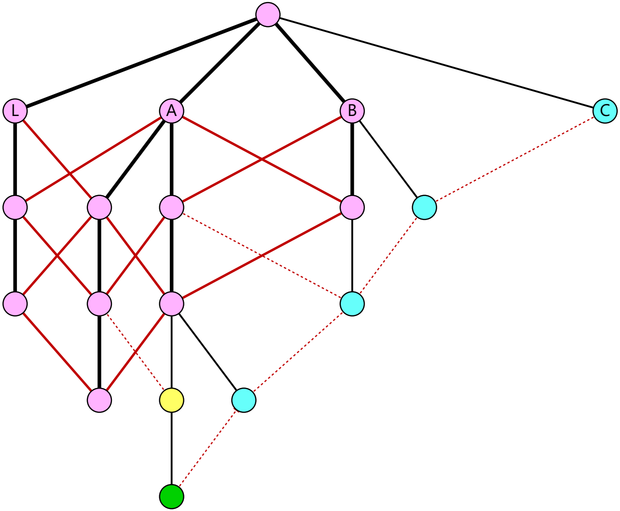
# Updating the view

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



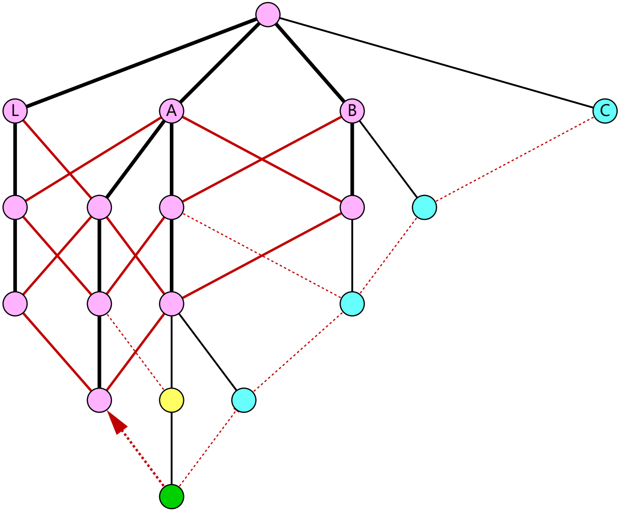
# Updating the view

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



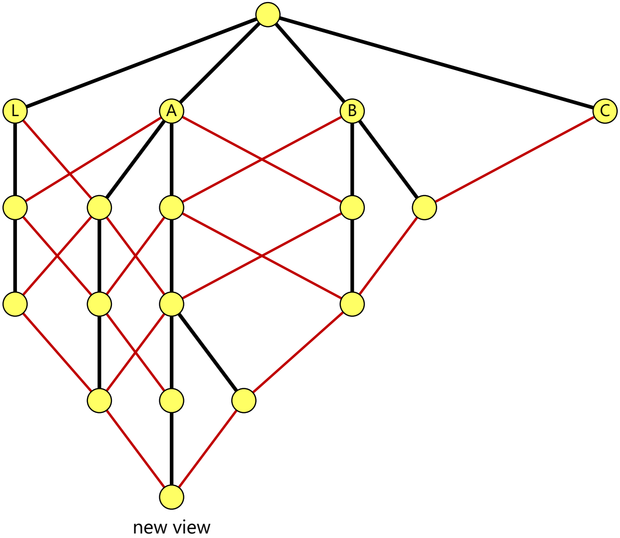
# Updating the view

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



# Updating the view

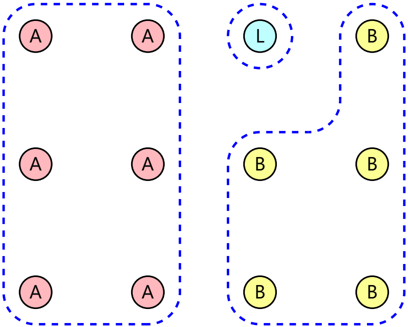
An agent updates its internal state by *merging* its view with the views it receives from its neighbors.



# Computability
# (with a Unique Leader)

# General computation

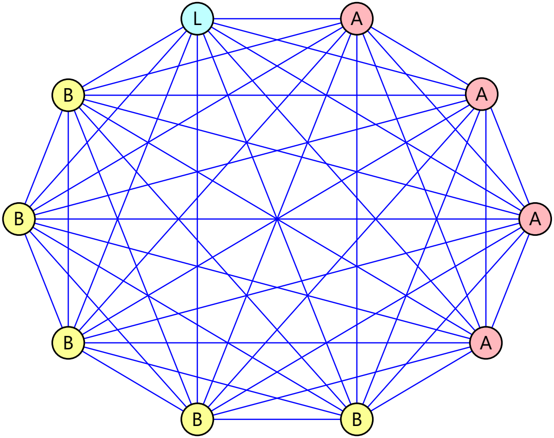
In general, we may assume that each agent has an *input* and has to compute an *output* depending on the entire network's inputs.




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 may be computable:

A function $f$ is multi-aggregate if it is of the form $f(x_p, \mu)$, where $x_p$ is the input of agent $p$ and $\mu$ is the multi-set of all inputs.

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

**Observation**

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

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

**Observation**

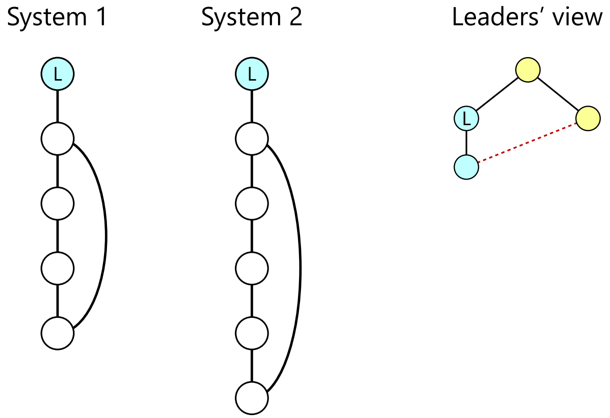
*If the Generalized Counting Problem is solvable in $f(n)$ rounds, then every multi-aggregate function is computable in $f(n)$ rounds.*

# Lower Bound on Counting (with a Unique Leader)

# Lower bound

**Theorem**

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

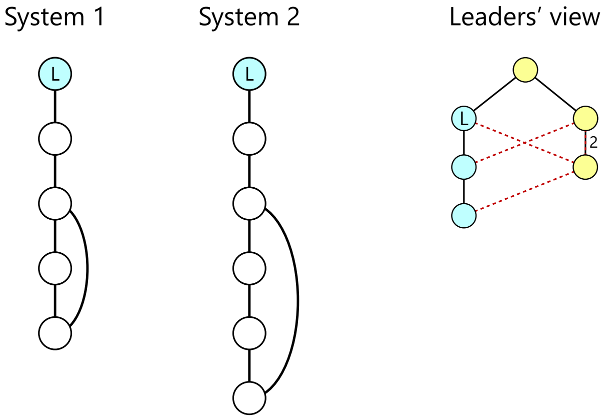



Round 1

# Lower bound

**Theorem**

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



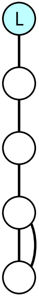

Round 2

# Lower bound

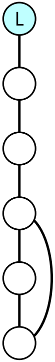
**Theorem**

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

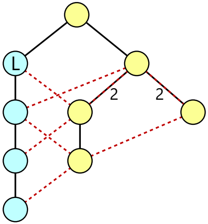
System 1




System 2




Leaders' view

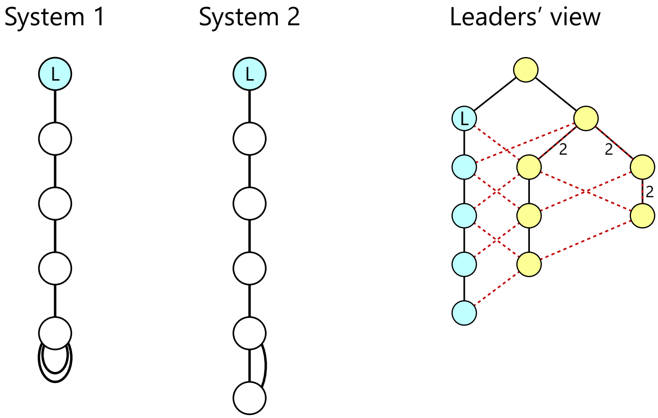



Round 3

# Lower bound

**Theorem**

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



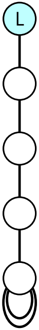

Round 4

# Lower bound

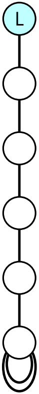
**Theorem**

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

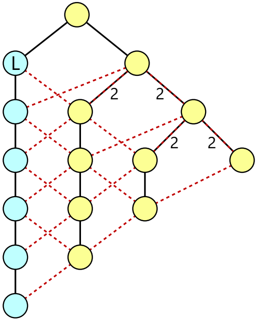
System 1




System 2




Leaders' view



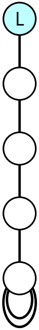

Round 5

# Lower bound

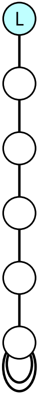
**Theorem**

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

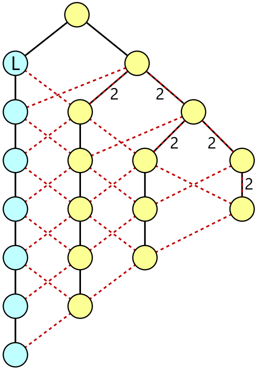
System 1




System 2




Leaders' view



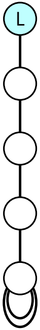

Round 6

# Lower bound

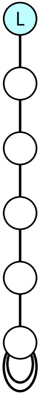
**Theorem**

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

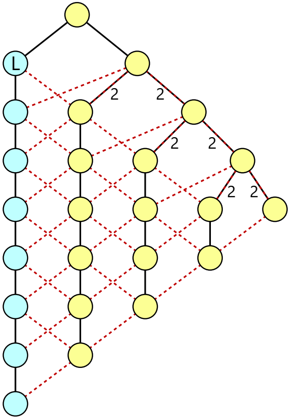
System 1




System 2




Leaders' view



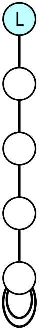

Round 7

# Lower bound

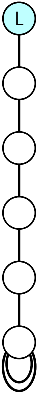
**Theorem**

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

System 1

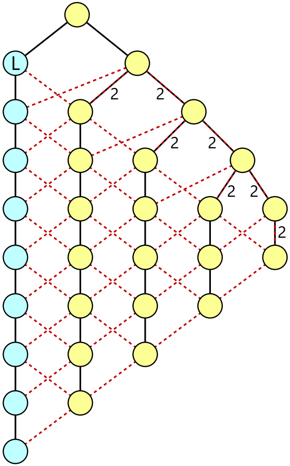



System 2




Round 8

Leaders' view

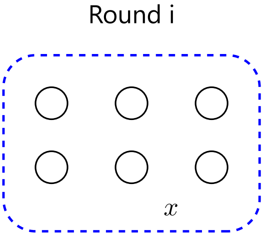


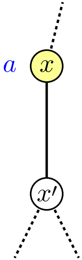
# Stabilizing Computation with a Unique Leader

# 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.

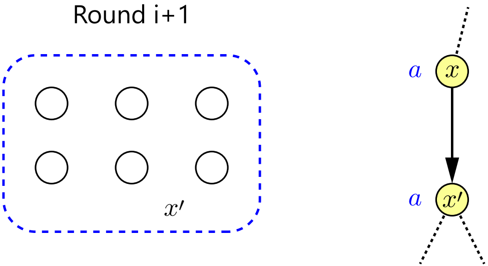


# Computing anonymities

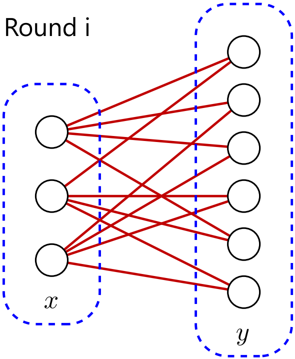
Suppose we know the anonymity of a node $x$ in the history tree.

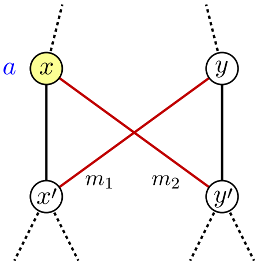
If $x$ has only one child $x'$, then $x'$ must have the same anonymity.



# Computing anonymities

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

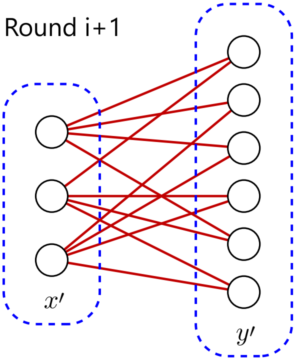


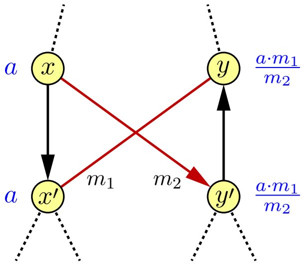




If the agents represented by $x$ have observed agents whose corresponding node $y$ has only one child $y'$, then we can compute the anonymity of $y$ and $y'$, as well.

# Computing anonymities

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

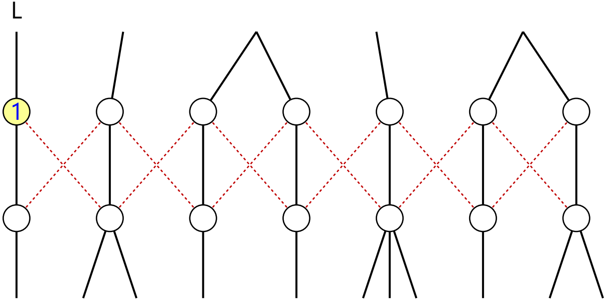



If the agents represented by $x$ have observed agents whose corresponding node $y$ has only one child $y'$, then we can compute the anonymity of $y$ and $y'$, as well.

# 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).

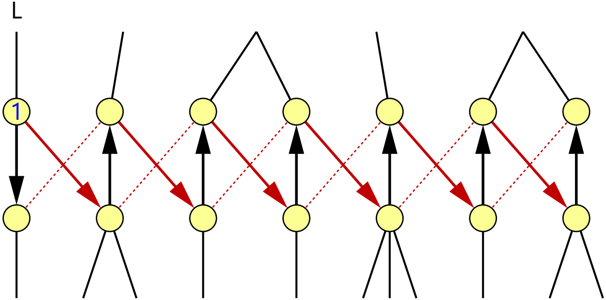



Since there are $n$ agents, the tree can branch at most $n-1$ times.

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).



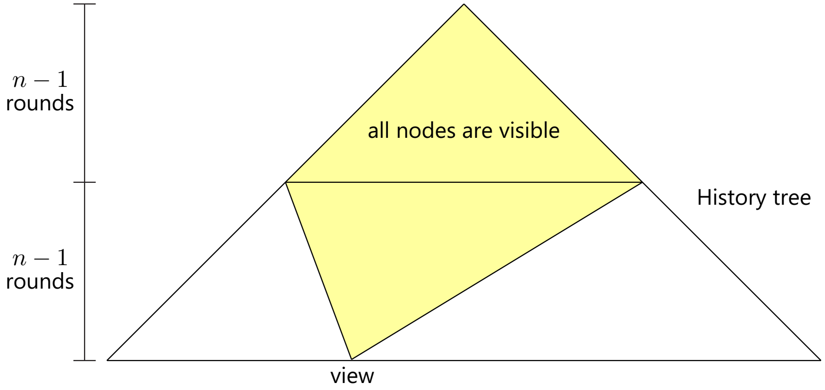

Since there are $n$ agents, the tree can branch at most $n-1$ times.

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**

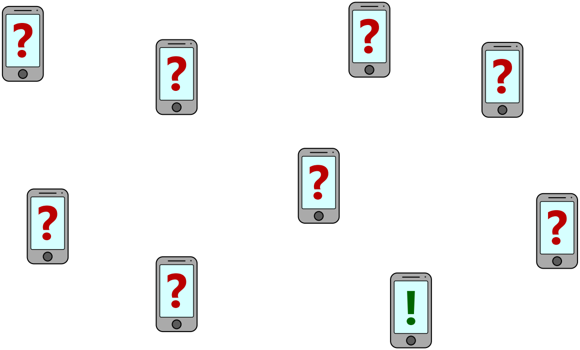
*The Generalized Counting Problem can be stably solved in $2n-2$ rounds (without explicit termination).*




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

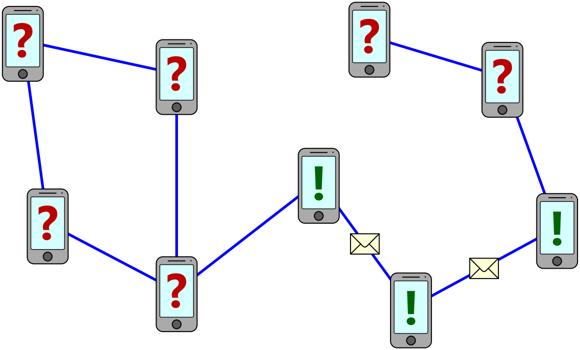
If the network is connected at all rounds, every news reaches every agent in at most $n-1$ rounds.




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

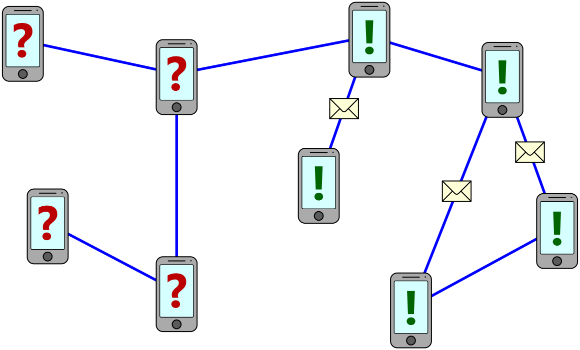
If the network is connected at all rounds, every news reaches every agent in at most $n-1$ rounds.




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

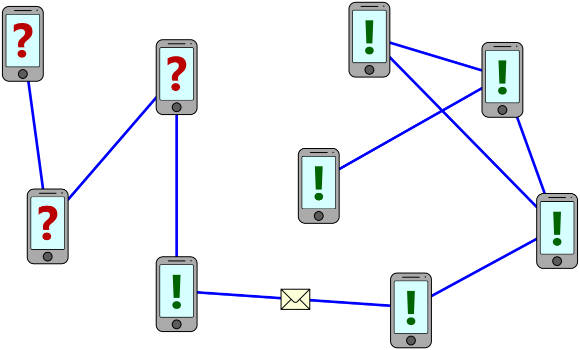
If the network is connected at all rounds, every news reaches every agent in at most $n-1$ rounds.




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

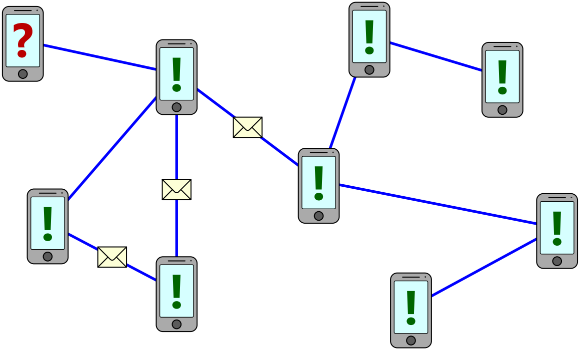
If the network is connected at all rounds, every news reaches every agent in at most $n-1$ rounds.




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

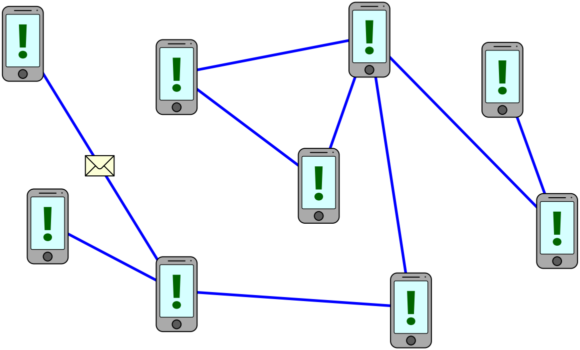
If the network is connected at all rounds, every news reaches every agent in at most $n-1$ rounds.




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).

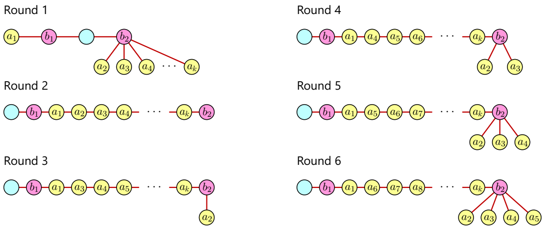
# Propagation of information

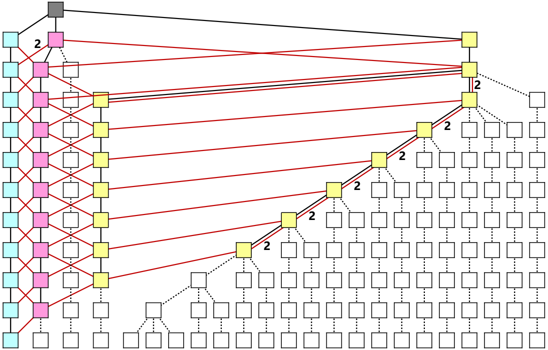
If the network is connected at all rounds, every news reaches every agent in at most $n - 1$ rounds.



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).

# Counterexample for termination

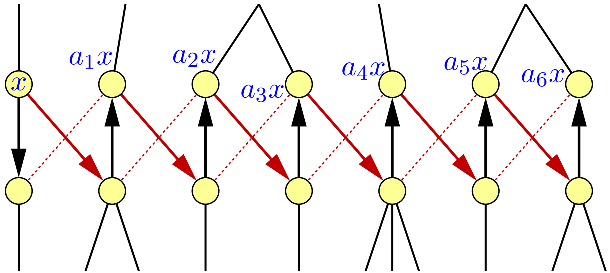


# Leaderless Computation

# 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 $2n$ 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.

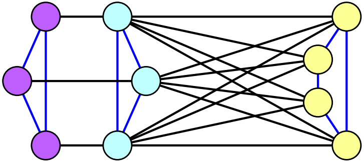
Examples include the mean (cf. *Average Consensus Problem*), variance, median, maximum, mode, and other statistical functions.

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

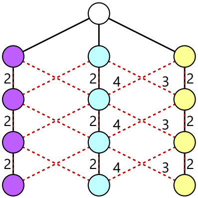
# Leaderless computation

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

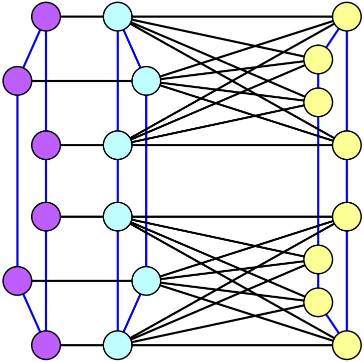
System 1



History tree

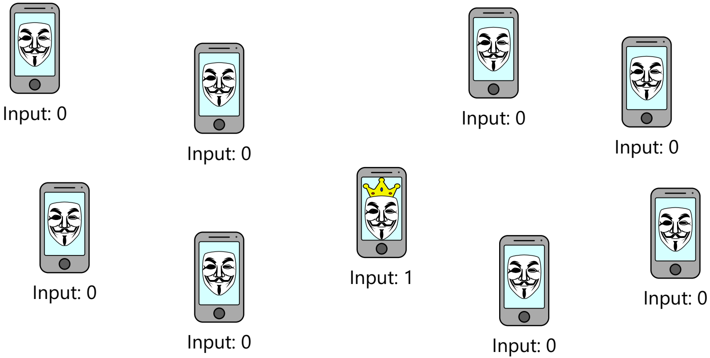



System 2



# Leaderless computation

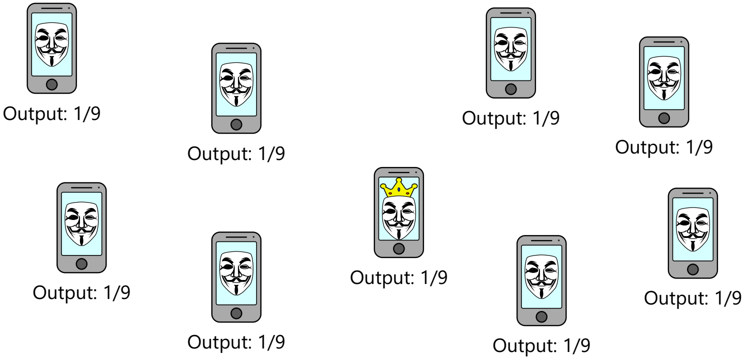
Also, $2n$ is a lower bound for the Average Consensus Problem.




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, $2n$ 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.

# Conclusion

**Theorem (Di Luna–V., 2022–2023)**

*Any function that is computable in connected anonymous dynamic networks with no Leader can be computed:*

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

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

**Theorem (Di Luna–V., 2022–2023)**

*Any function that is computable in connected anonymous dynamic networks with a unique Leader can be computed:*

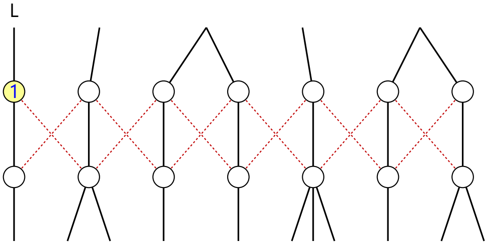
- *in $2n$ rounds without explicit termination,*
- *in $3n$ rounds with termination.* **(Tomorrow's seminar!)**

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

# Leader Election with Knowledge of $n$

# Leader election

We can elect a unique leader in a network with knowledge of $n$ if and only if there is a non-branching level in the history tree where at least one node has anonymity $1$.




- In a static network, this algorithm terminates in $2n$ rounds and either elects a leader or reports that no leader can be elected.
- In a dynamic network, it may be necessary to wait indefinitely for a node of anonymity $1$ to be created (stabilizing).

$\Longrightarrow$ Counting and Leader election are equivalent!