Advances in Anonymous Dynamic Networks

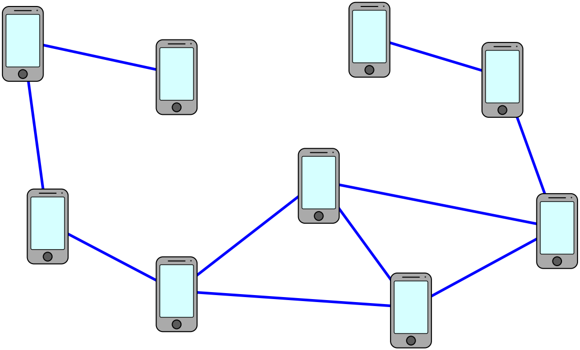
Giovanni Viglietta

Joint work with Giuseppe A. Di Luna

JAIST – June 29, 2022

Dynamic networks

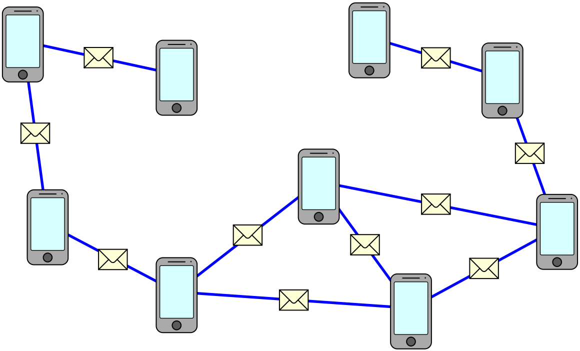
In a *dynamic network*, some machines (or agents) are connected with each other through links that may change over time.



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

Dynamic networks

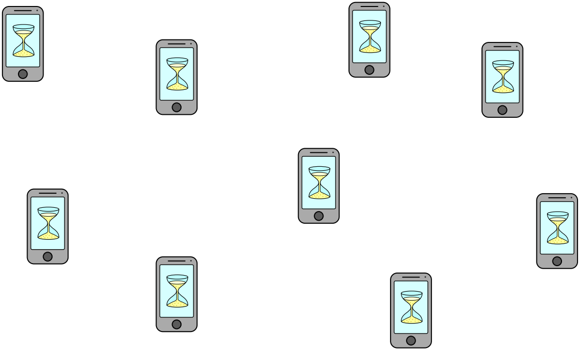
In a *dynamic network*, some machines (or agents) are connected with each other through links that may change over time.



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

Dynamic networks

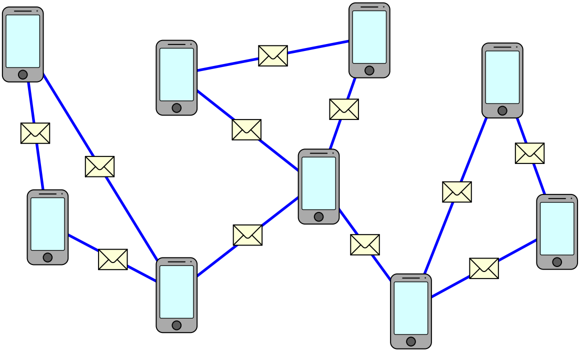
In a *dynamic network*, some machines (or agents) are connected with each other through links that may change over time.



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

Dynamic networks

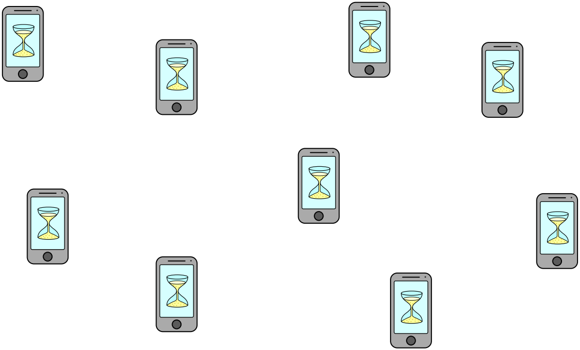
In a *dynamic network*, some machines (or agents) are connected with each other through links that may change over time.



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

Dynamic networks

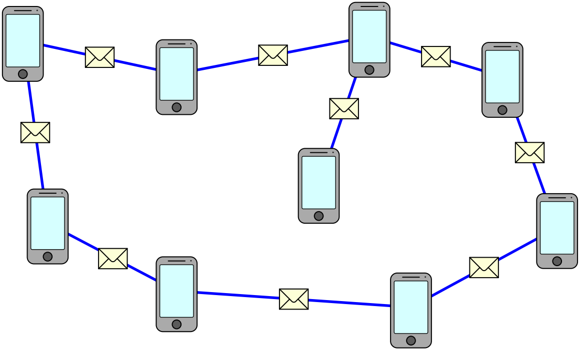
In a *dynamic network*, some machines (or agents) are connected with each other through links that may change over time.



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

Dynamic networks

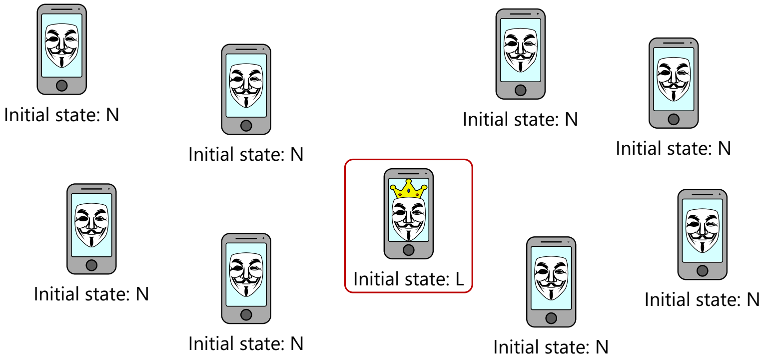
In a *dynamic network*, some machines (or agents) are connected with each other through links that may change over time.



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

Counting anonymous agents with a Leader

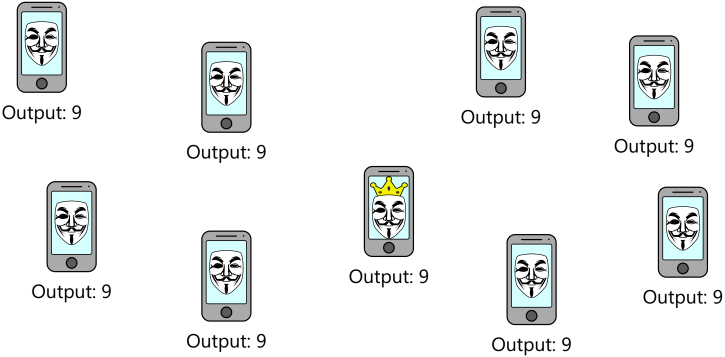
A typical assumption is that the dynamic network is *anonymous*, i.e., all agents start with the same state, except one: the *Leader*.



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

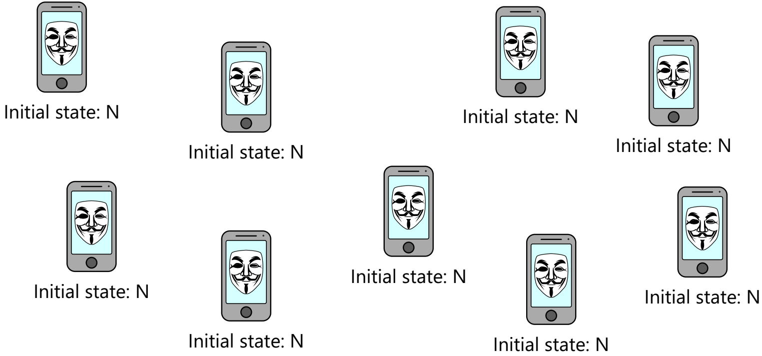
A typical assumption is that the dynamic network is *anonymous*, i.e., all agents start with the same state, except one: the *Leader*.



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.)

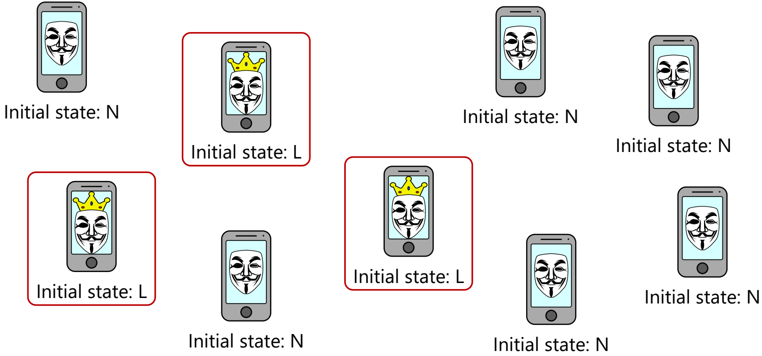
No Leader or multiple Leaders

It is also interesting to explore the scenario where *no Leader* or *multiple Leaders* are present.



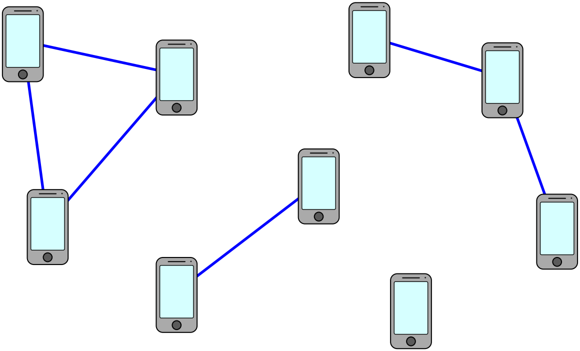
No Leader or multiple Leaders

It is also interesting to explore the scenario where *no Leader* or *multiple Leaders* are present.



Disconnected networks

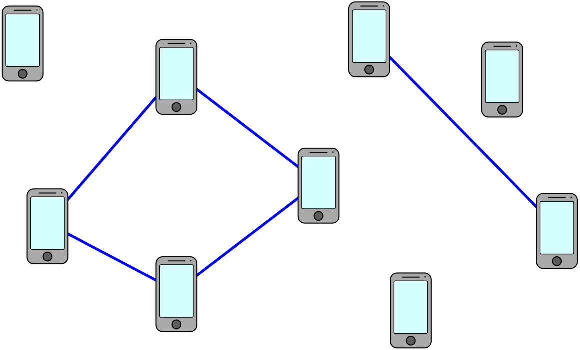
Also, while the network is typically assumed to be connected at every round, we may relax this assumption.



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

Disconnected networks

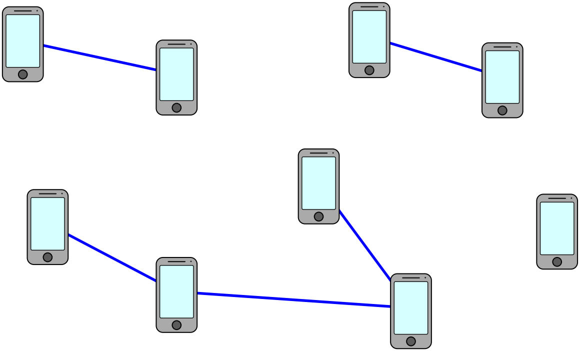
Also, while the network is typically assumed to be connected at every round, we may relax this assumption.



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

Disconnected networks

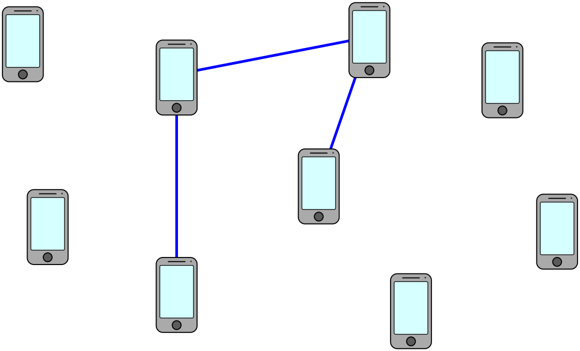
Also, while the network is typically assumed to be connected at every round, we may relax this assumption.



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

Disconnected networks

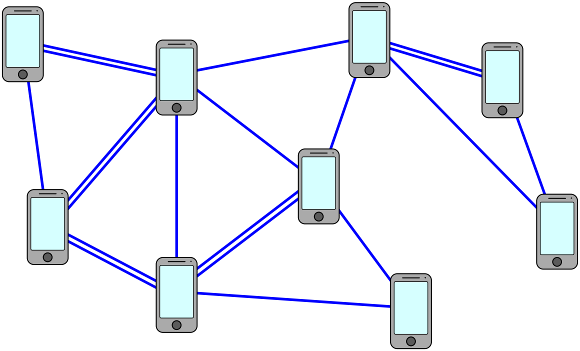
Also, while the network is typically assumed to be connected at every round, we may relax this assumption.



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

Previous work 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–Viglietta:** $3n$ rounds (FOCS 2022)
- **Di Luna–Viglietta:** $(\ell^2 + \ell + 1)Tn$ rounds (Today's talk)

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)

Our previous results

Theorem

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

- *Stabilizing algorithm in $2n$ rounds.*
- *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.

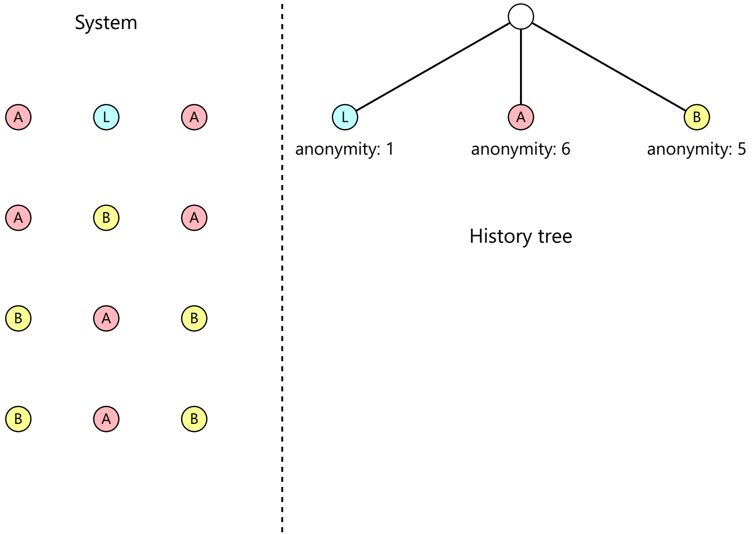
The theorem applies not only to the Counting Problem, but to *all* problems computable in anonymous (dynamic) networks.

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.

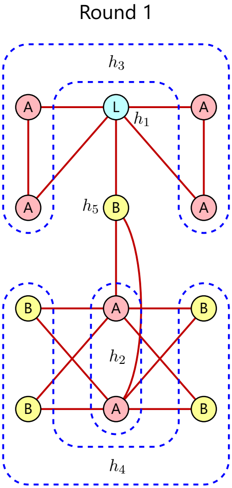
History tree

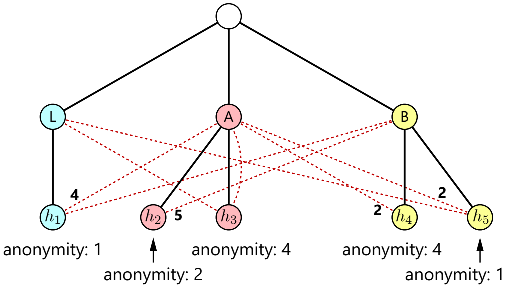
We introduced the *history tree* as our main tool of investigation.



History tree

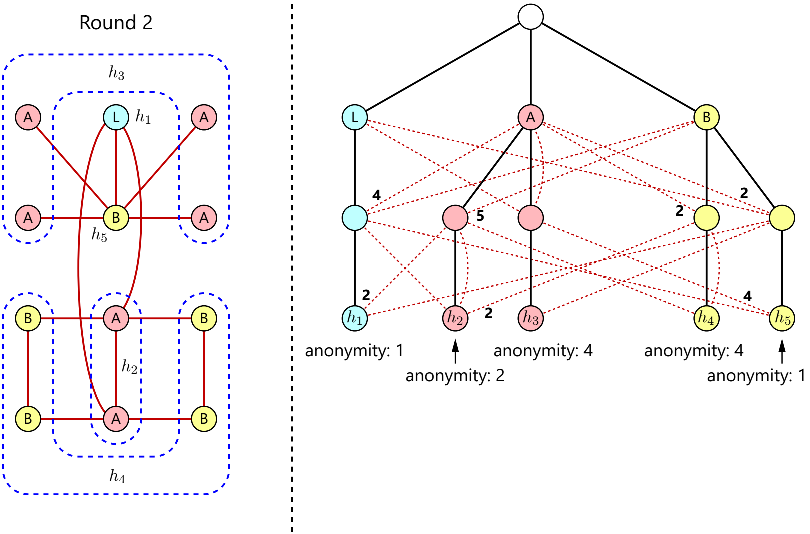
We introduced the *history tree* as our main tool of investigation.





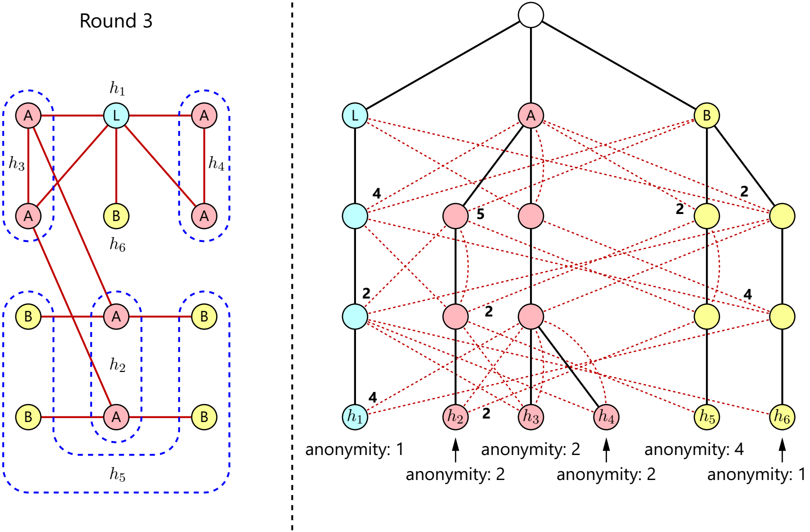
History tree

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



History tree

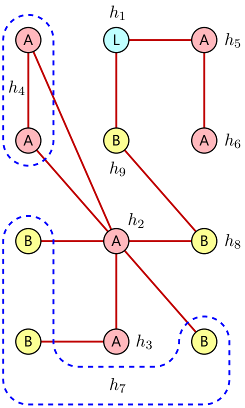
We introduced the *history tree* as our main tool of investigation.

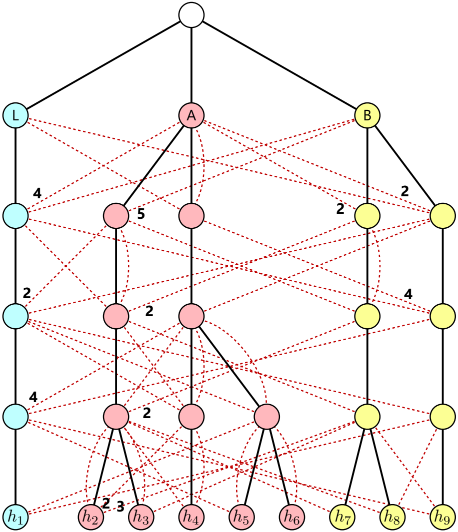


History tree

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

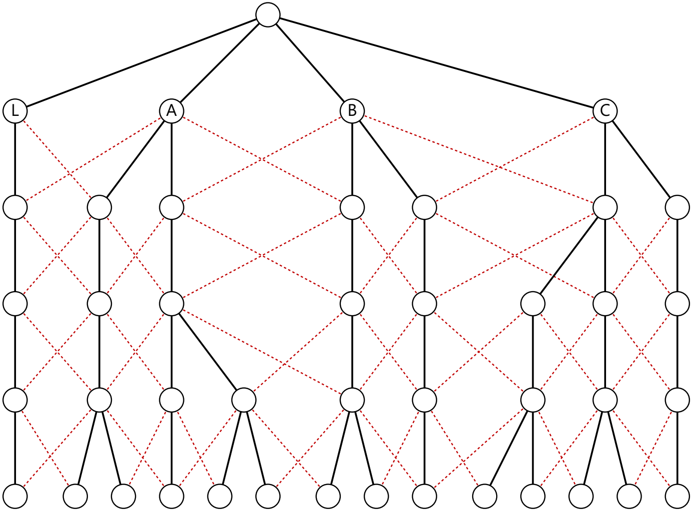
Round 4





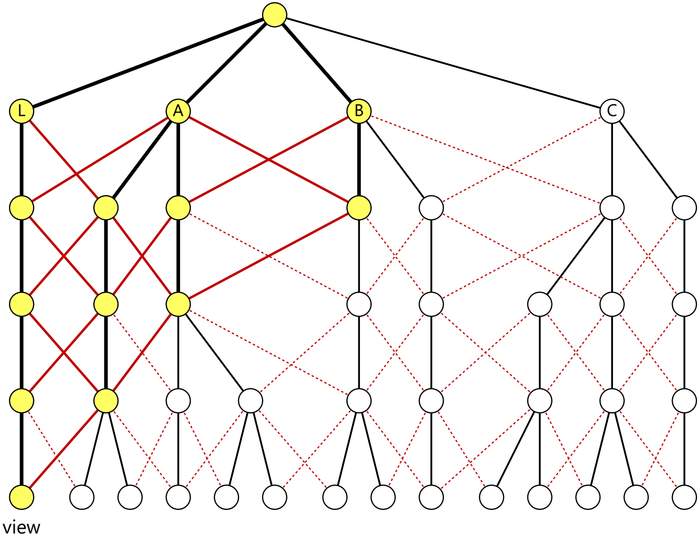
View of a history tree

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



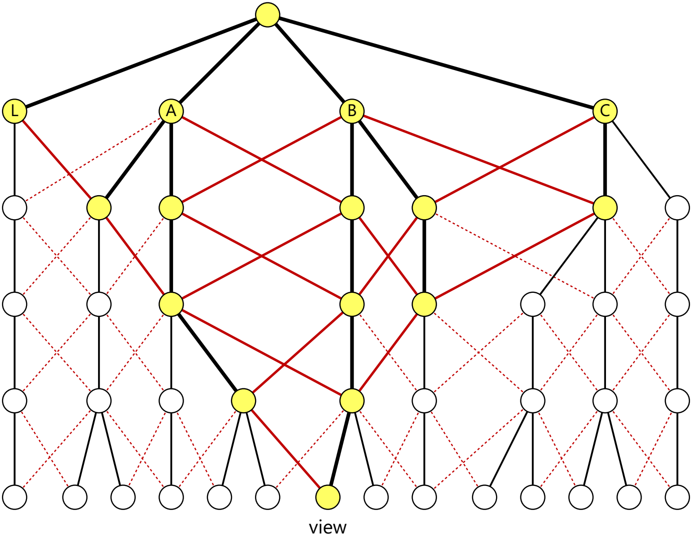
View of a history tree

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.

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.

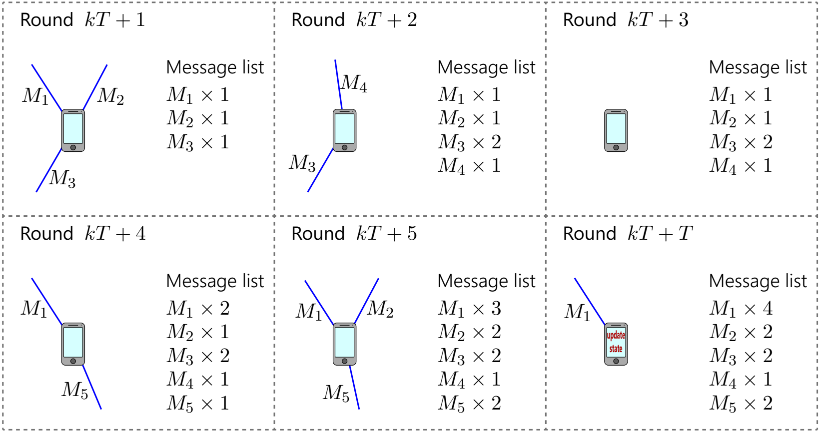
This is good because, at round i, the size of a view is only $O(i^3)$.

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.

T-time-connected networks

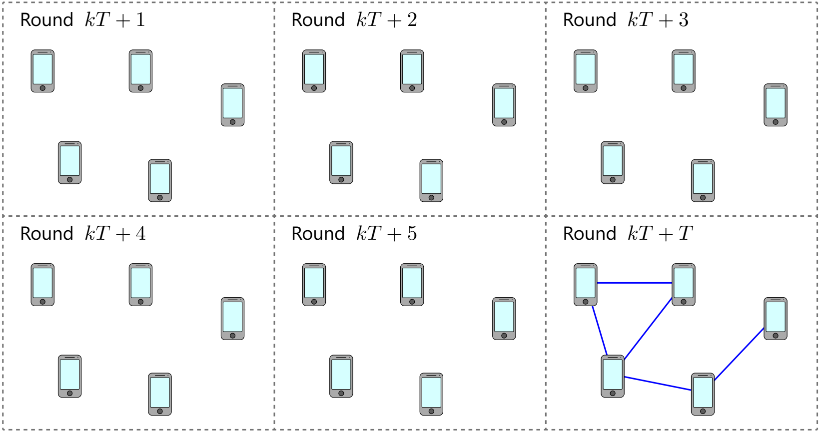
Any algorithm for $T = 1$ can be adapted to networks with $T > 1$, assuming T is known by all agents.



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

T-time-connected networks

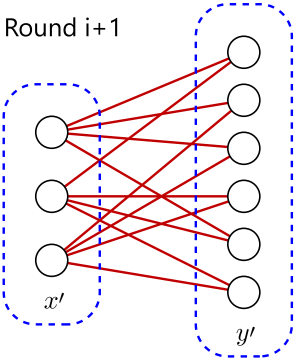
This is the best we can do: Consider, for instance, a network that contains no links for $T-1$ out of every T rounds.

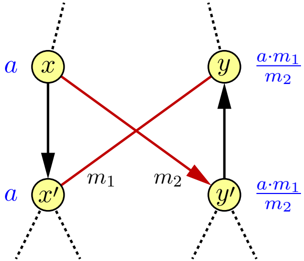


Thus, the Counting Problem has a lower bound of $2Tn$ rounds.

Leaderless computation

We will re-use a technique from our previous stabilizing algorithm. Suppose we know the anonymity of a node x with a single child x'.

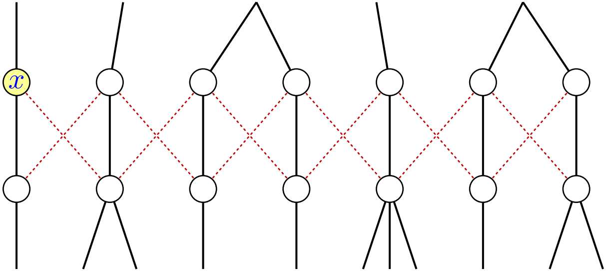




If the agents represented by x have observed agents whose corresponding node y has only one child y', then we can compute the anonymity of y and y', as well.

Leaderless computation

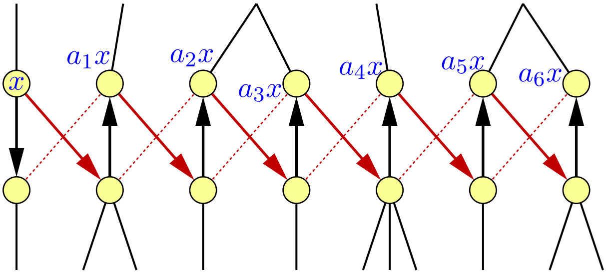
If all nodes in a level have only one child, we can compute the anonymity of each one of them as a function of a single node's anonymity x.



Since there are only n agents, the tree can branch at most n times.

Thus, among the first n levels, there must be a level where no node branches. In this level, we can compute all anonymities up to a common factor x.

If all nodes in a level have only one child, we can compute the anonymity of each one of them as a function of a single node's anonymity x.

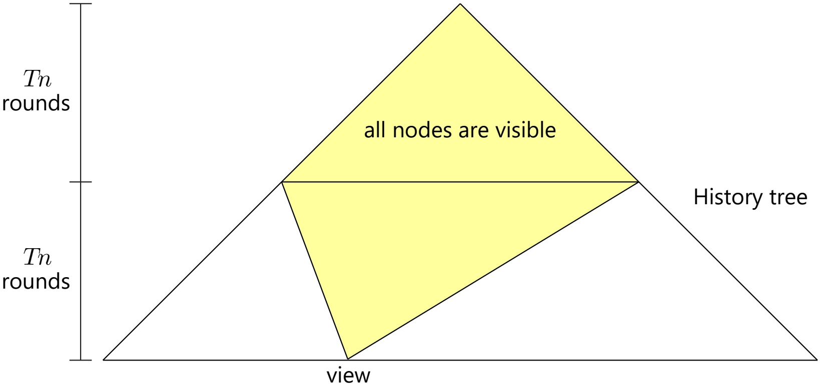


Since there are only n agents, the tree can branch at most n times.

Thus, among the first n levels, there must be a level where no node branches. In this level, we can compute all anonymities up to a common factor x.

Leaderless computation

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



Thus, within $2Tn$ rounds, all nodes can count all agents with any given input up to a common factor 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 *ratio-multi-aggregate* functions.

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

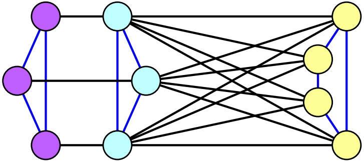
More specifically, we can compute all ratio-multi-aggregate functions:

- in $2Tn$ rounds without explicit termination;
- in $T(n + N)$ with termination, if an upper bound $N \geq n$ is known by all agents (waiting TN rounds yields a certificate).

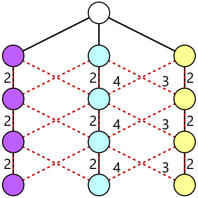
Leaderless computation

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.

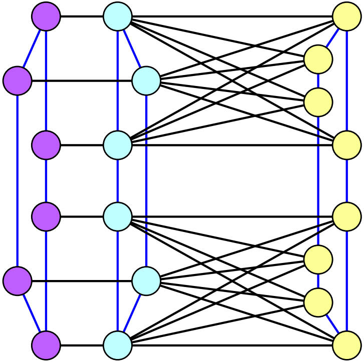
System 1



History tree

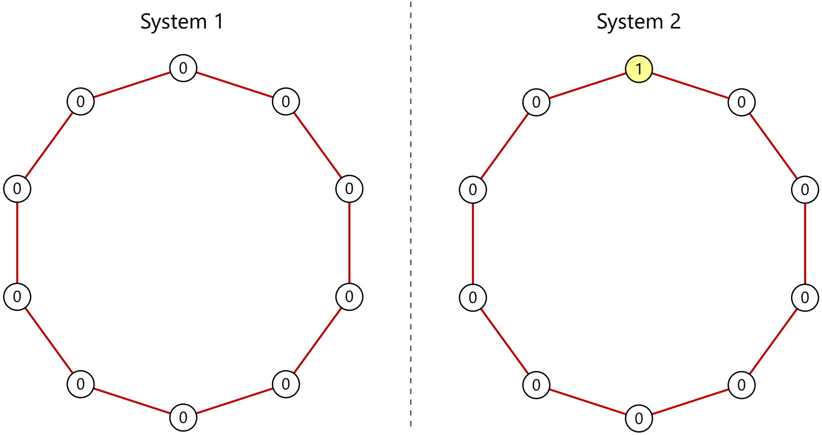


System 2



Leaderless computation

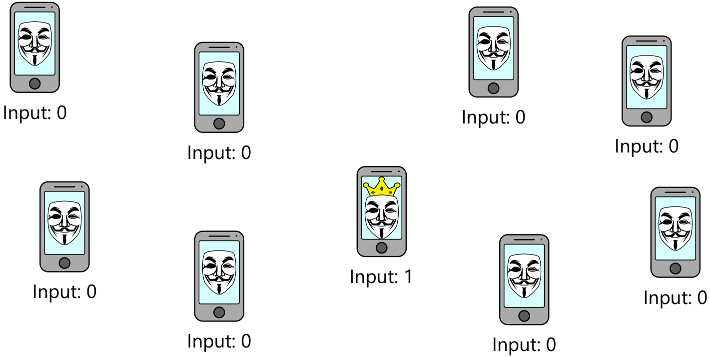
Note that termination is impossible if nothing is known about n.



If an algorithm takes m rounds to terminate on a ring with all 0's (note that m does not depend on n), then it terminates incorrectly on a ring of size $2m + 2$ with all 0's and a single 1.

Leaderless computation

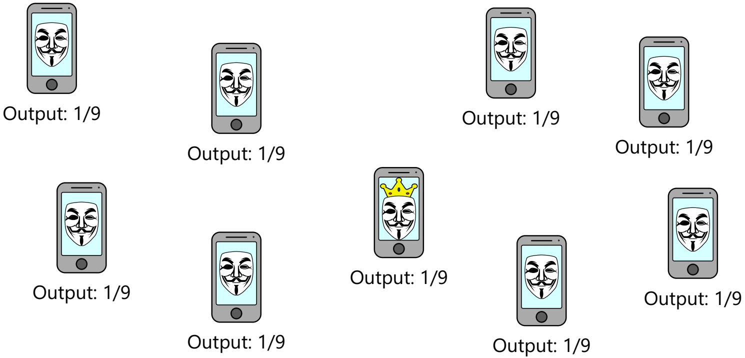
Also, $2Tn$ is a lower bound on the Average Consensus Problem.



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

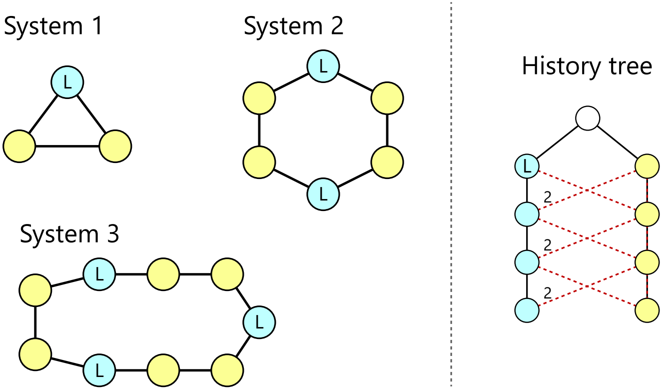
Also, $2Tn$ is a lower bound on the Average Consensus Problem.



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.

Multiple Leaders

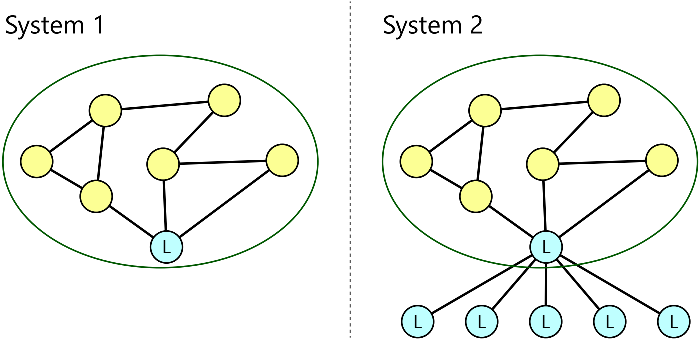
The Counting Problem is unsolvable with no knowledge on the number of Leaders, ℓ.



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

Multiple Leaders

Having more than one Leader may not be very helpful. Actually, multiple Leaders introduce more symmetry in the network.

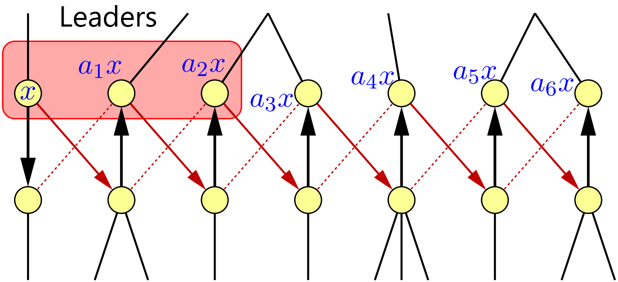


If multiple Leaders were always helpful, a single Leader could "pretend" to see several other Leaders to speed up computation.

Multiple Leaders

For a stabilizing (non-terminating) Counting algorithm, we can use the same technique as before (assuming that all agents know ℓ).

As soon as no node branches for a round, we can compute all anonymities as a function of a single Leader node's anonymity, x.

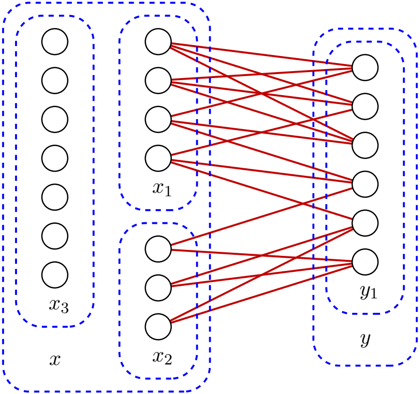


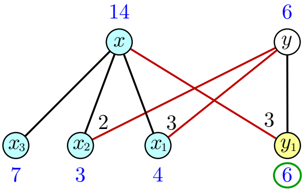
We know that $x + a_1x + a_2x = \ell$, and we can determine x.

This yields a non-terminating Counting algorithm that stabilizes in at most $2Tn$ rounds (optimal).

Single-Leader algorithm summary

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.



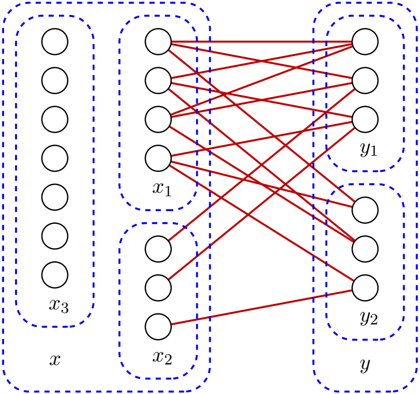


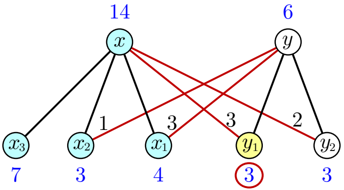
Guess on y_1: $\dfrac{4 \cdot 3 + 3 \cdot 2}{3} = 6$

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

Single-Leader algorithm summary

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.



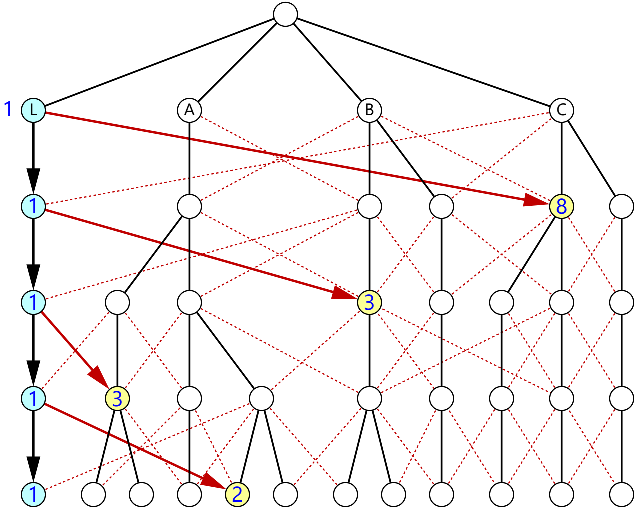


Guess on y_1: $\dfrac{4 \cdot 3 + 3 \cdot 1}{3} = 5$

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

Single-Leader algorithm summary

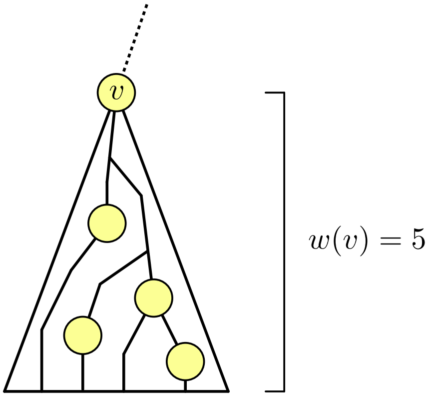
We can make one guess per round using the Leader's observations.



How do we know which guesses are correct?

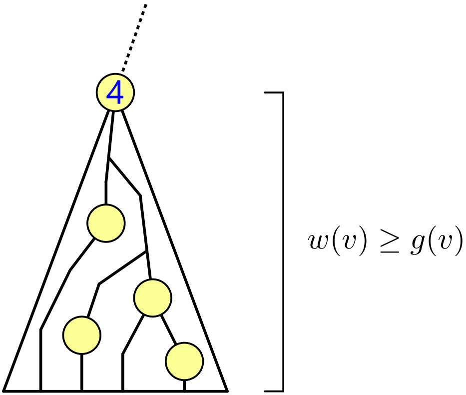
Single-Leader algorithm summary

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.



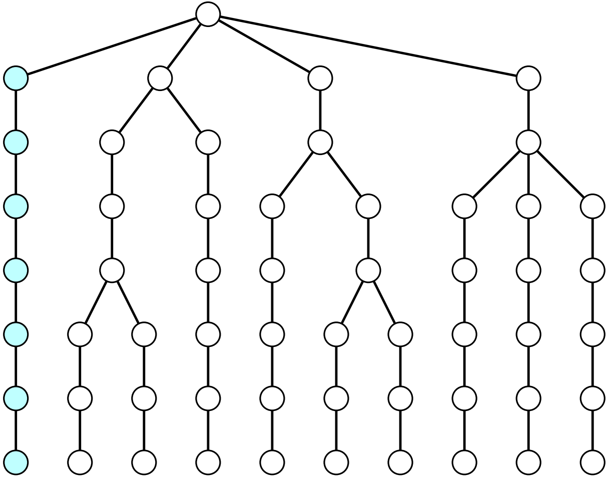
Single-Leader algorithm summary

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



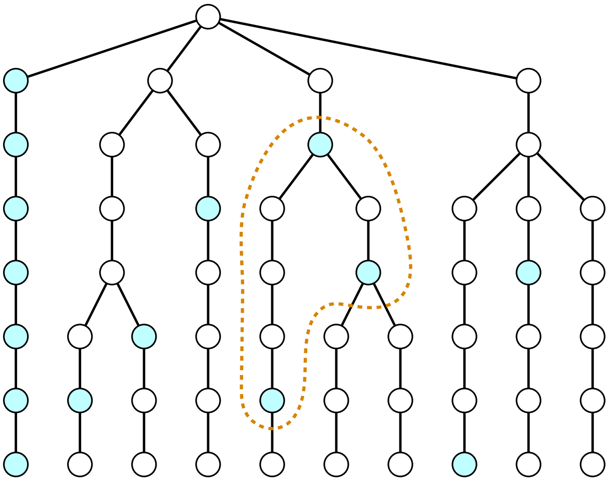
Correctness Criterion: If a node v is heavy and no descendants of v are heavy, then the guess on v is correct.

Single-Leader algorithm summary



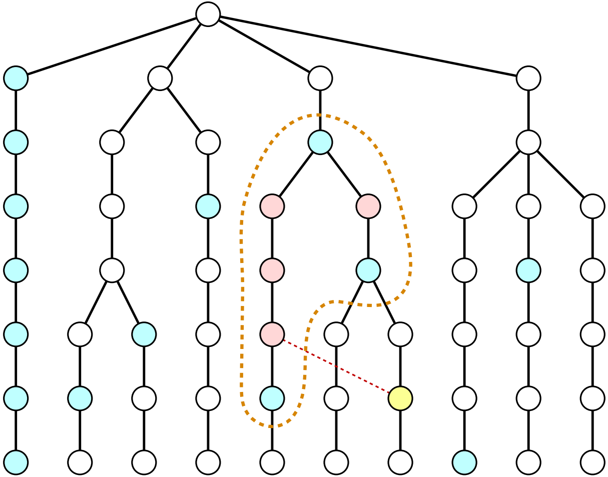
Initially, all Leader nodes are Guessers with anonymity $\ell = 1$.
Eventually, some guessed nodes become correct.

Single-Leader algorithm summary



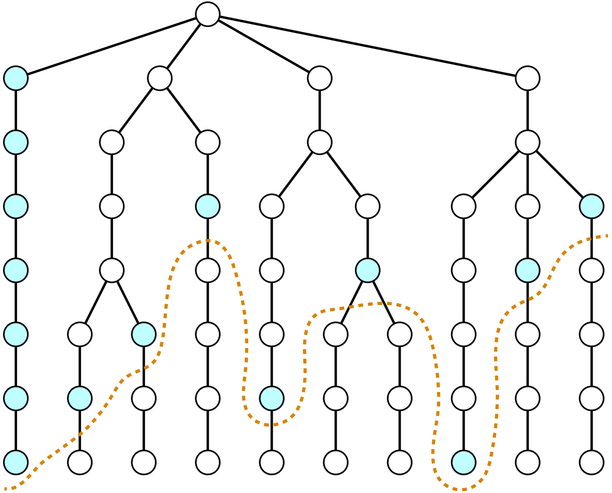
Correct nodes will form *isles*, which allow us to determine more anonymities.

Single-Leader algorithm summary



In turn, all nodes whose anonymity is known and whose children's anonymities are known become new Guessers.

Single-Leader algorithm summary

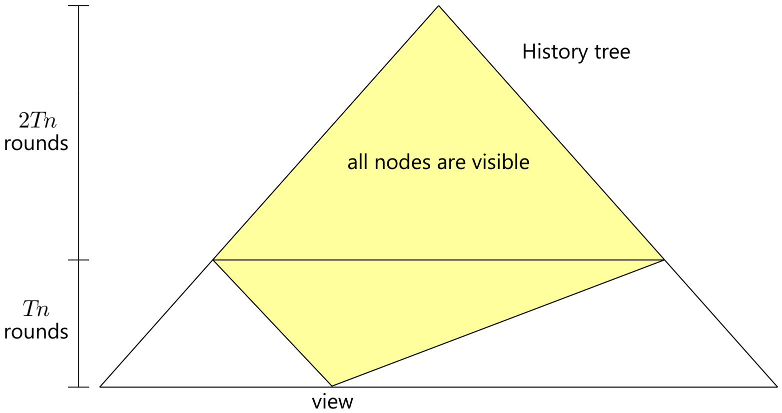


Eventually, some nodes determine a *cut* of the history tree, in which case we have an estimate n' on n, given by their sum.

Single-Leader algorithm summary

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

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

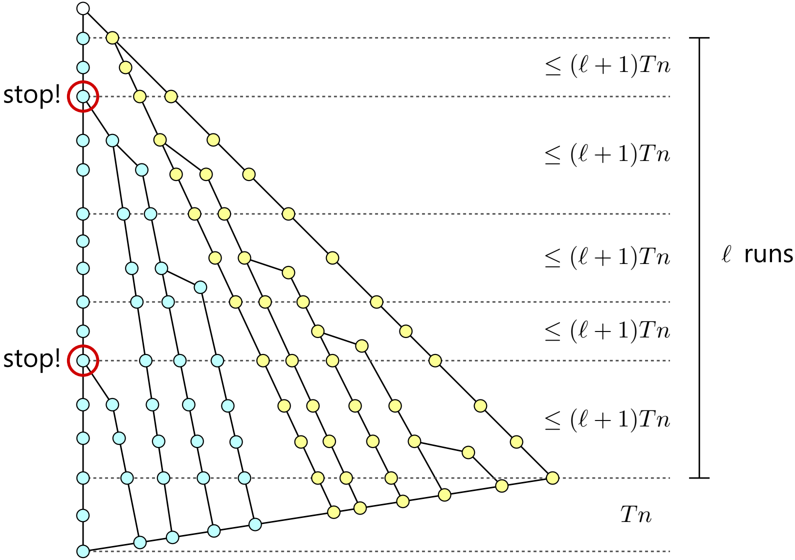


If $n' = n$, then no new nodes appear, and the algorithm terminates.

This happens within $3Tn$ rounds.

Multiple Leaders

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



Multiple Leaders

Terminating algorithm for $\ell > 1$ Leaders:

- Do ℓ runs of the terminating algorithm for $\ell = 1$ as follows.
  - Choose a branch of Leader nodes and assign it anonymity x.
  - Run the terminating algorithm assuming $x = \ell$ (note that all guesses are still upper bounds of the real anonymities).
  - If we encounter a node where the chosen Leader branch splits, stop the current run and proceed with the next.
  - Else, the algorithm eventually terminates with an estimate n'_i of n, as well as an estimate of all nodes' anonymities.
  - If the sum of Leaders' anonymities is ℓ, store n'_i and proceed with the next run. Else, repeat this run with $x = \ell - 1$, etc.
- We end up with at most ℓ estimates of n: n'_1, n'_2, ...
  Wait another $T \cdot \max\{n'_1, n'_2, \dots\}$ rounds to confirm them.
- If the n'_i's have not changed, they are all equal to n and all correct (note that at least one run must be correct, because the Leaders can split at most $\ell - 1$ times).

Conclusions

Theorem

Any problem that is solvable in a T-time-connected 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

Any problem that is solvable in a T-time-connected 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.

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