Optimal Terminating Computation in Anonymous Dynamic Networks with a Leader

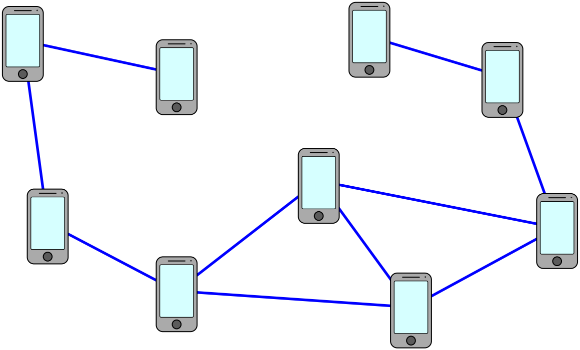
Giovanni Viglietta

Joint work with Giuseppe A. Di Luna

JAIST – December 13, 2021

Dynamic networks

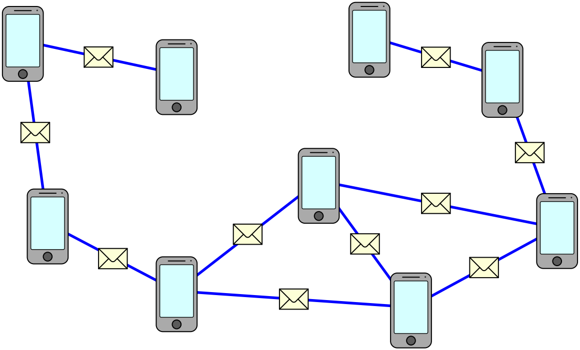
In a *dynamic network*, some machines (or agents) are connected with each other through links that may change over time.



Assume that, at every *round*, the links form a connected graph.
What can be computed by this network, and in how many rounds?

Dynamic networks

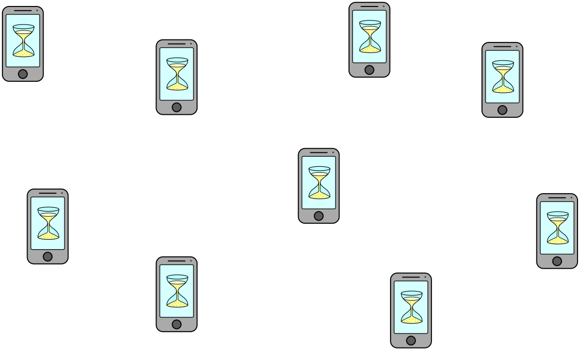
In a *dynamic network*, some machines (or agents) are connected with each other through links that may change over time.



Assume that, at every *round*, the links form a connected graph. What can be computed by this network, and in how many rounds?

Dynamic networks

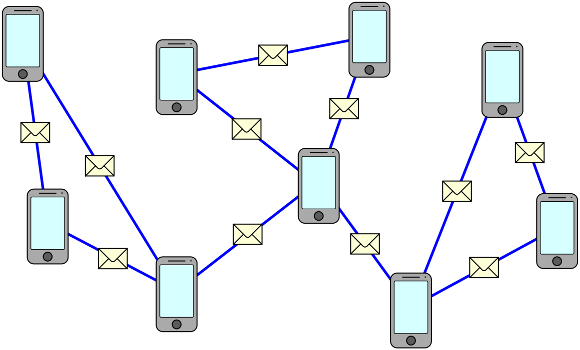
In a *dynamic network*, some machines (or agents) are connected with each other through links that may change over time.



Assume that, at every *round*, the links form a connected graph.
What can be computed by this network, and in how many rounds?

Dynamic networks

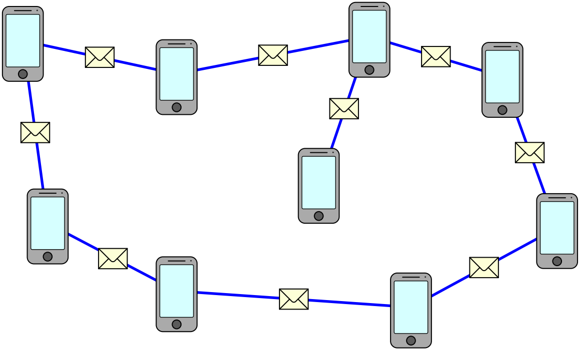
In a *dynamic network*, some machines (or agents) are connected with each other through links that may change over time.



Assume that, at every *round*, the links form a connected graph. What can be computed by this network, and in how many rounds?

Dynamic networks

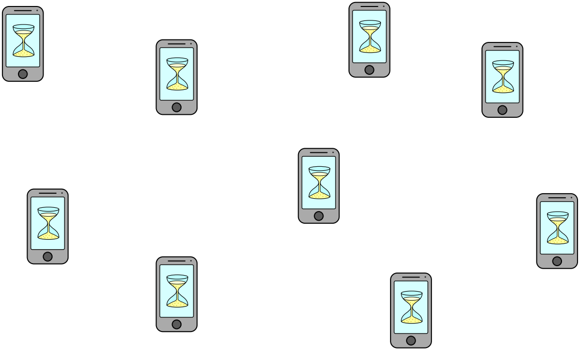
In a *dynamic network*, some machines (or agents) are connected with each other through links that may change over time.



Assume that, at every *round*, the links form a connected graph. What can be computed by this network, and in how many rounds?

Dynamic networks

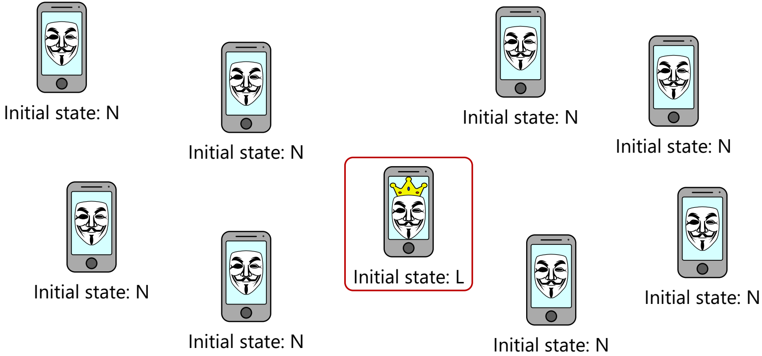
In a *dynamic network*, some machines (or agents) are connected with each other through links that may change over time.



Assume that, at every *round*, the links form a connected graph.
What can be computed by this network, and in how many rounds?

Counting anonymous agents with a Leader

We assume the dynamic network to be *anonymous*, i.e., all agents start with the same internal state, except one: the *Leader*.

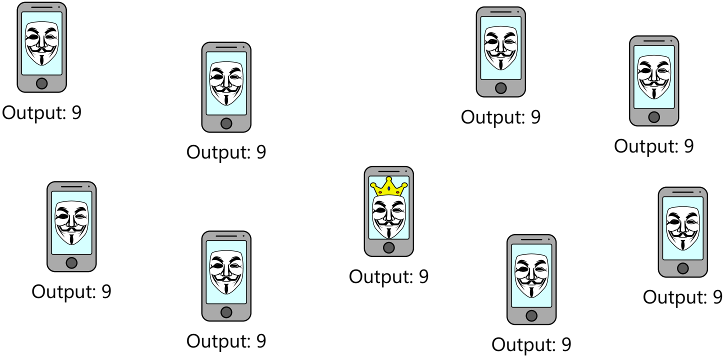


Counting Problem: Eventually, all agents must know the total number of agents, n. Is it possible? In how many rounds at most?

Note: Knowing n allows agents to solve a large class of problems.

Counting anonymous agents with a Leader

We assume the dynamic network to be *anonymous*, i.e., all agents start with the same internal state, except one: the *Leader*.



Counting Problem: Eventually, all agents must know the total number of agents, n. Is it possible? In how many rounds at most?

Note: Knowing n allows agents to solve a large class of problems.

Previous work

Theorem (Michail et al., SSS 2013)

In a static anonymous network,

1. *Without a Leader, counting processes is impossible.*

2. *With a unique Leader, counting can be done in* $\mathbf{2n}$ **rounds**.

Conjecture. *Counting in a dynamic network is impossible even with a Leader.*

Theorem (Di Luna et al., ICDCN 2014)

In a dynamic anonymous network with a unique Leader, counting agents can be done in an **exponential number of rounds**, *provided that an upper bound on* n *is known.*

Theorem (Di Luna–Baldoni, OPODIS 2015)

In a dynamic anonymous network with a unique Leader, counting agents can be done in an **exponential number of rounds**.

Theorem (Kowalski–Mosteiro, ICALP 2018, Best Paper Award)

In a dynamic anonymous network with a unique Leader, counting agents can be done in $\mathbf{O(n^4 \log^3 n)}$ **rounds**.

Previous work

Theorem (Michail et al., SSS 2013)

In a static anonymous network,

1. *Without a Leader, counting processes is impossible.*

2. *With a unique Leader, counting can be done in* $\mathbf{2n}$ **rounds**.

Conjecture. *Counting in a dynamic network is impossible even with a Leader.*

Theorem (Di Luna et al., ICDCN 2014)

In a dynamic anonymous network with a unique Leader, counting agents can be done in an **exponential number of rounds**, *provided that an upper bound on* n *is known.*

Theorem (Di Luna–Baldoni, OPODIS 2015)

In a dynamic anonymous network with a unique Leader, counting agents can be done in an **exponential number of rounds**.

Theorem (Kowalski–Mosteiro, ICALP 2018, Best Paper Award)

In a dynamic anonymous network with a unique Leader, counting agents can be done in $\mathbf{O(n^4 \log^3 n)}$ **rounds**. (Can we improve upon this?)

Previous seminar

The Counting Problem is "complete":

Observation (Folklore?)

If the Counting Problem is solvable in $f(n)$ rounds, then every (computable) function can be computed in $f(n)$ rounds.

Theorem (Previous seminar)

The Counting Problem can be solved in $2n - 2$ rounds in a connected anonymous dynamic network with a Leader, and is not solvable in less that $1.5n - 2$ rounds.

Open Problem: Note that agents' outputs only *stabilize* on the correct result. Is there a way for all agents to *know* when they have solved the problem and *terminate* in $O(n)$ rounds?

Previous seminar

The Counting Problem is "complete":

Observation (Folklore?)

If the Counting Problem is solvable in $f(n)$ rounds, then every (computable) function can be computed in $f(n)$ rounds.

Theorem (Previous seminar)

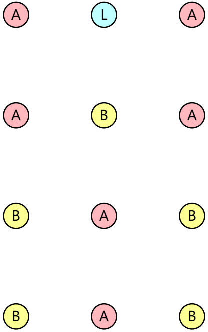
The Counting Problem can be solved in $2n-2$ rounds in a connected anonymous dynamic network with a Leader, and is not solvable in less that $1.5n-2$ rounds.

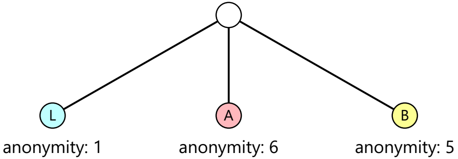
Open Problem: Note that agents' outputs only *stabilize* on the correct result. Is there a way for all agents to *know* when they have solved the problem and *terminate* in $O(n)$ rounds? (Today's talk)

History tree

We introduced the *history tree* as a tool for studying dynamic networks.

System

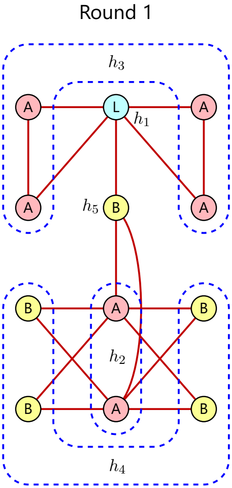


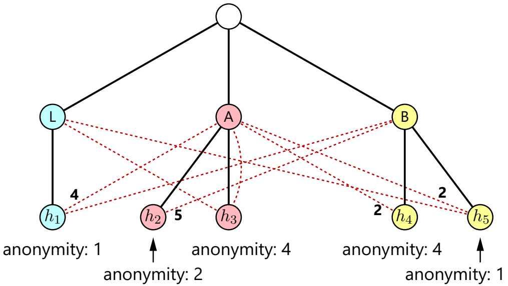


History tree

History tree

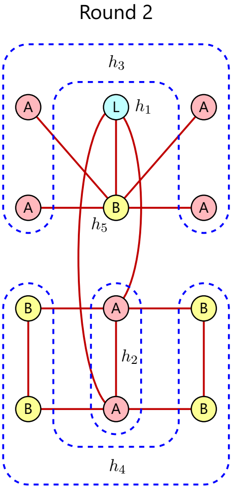
We introduced the *history tree* as a tool for studying dynamic networks.

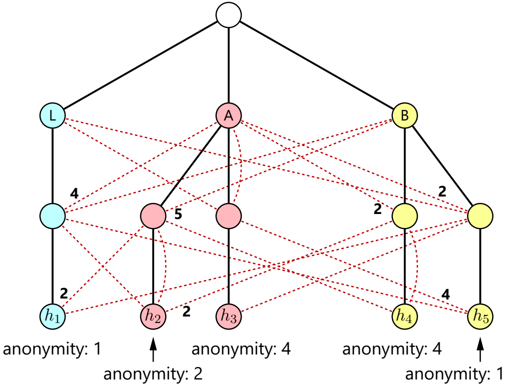




History tree

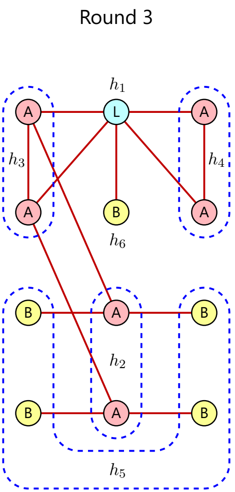
We introduced the *history tree* as a tool for studying dynamic networks.

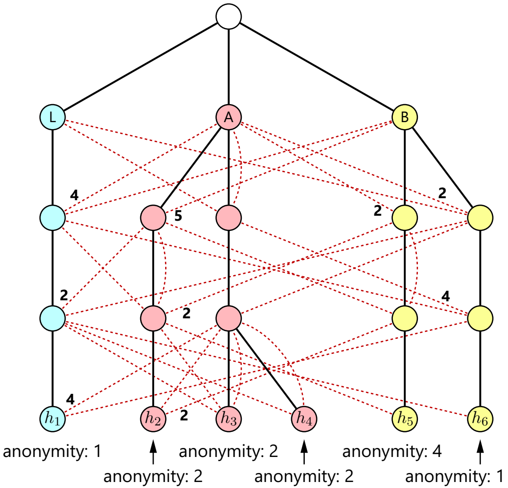




History tree

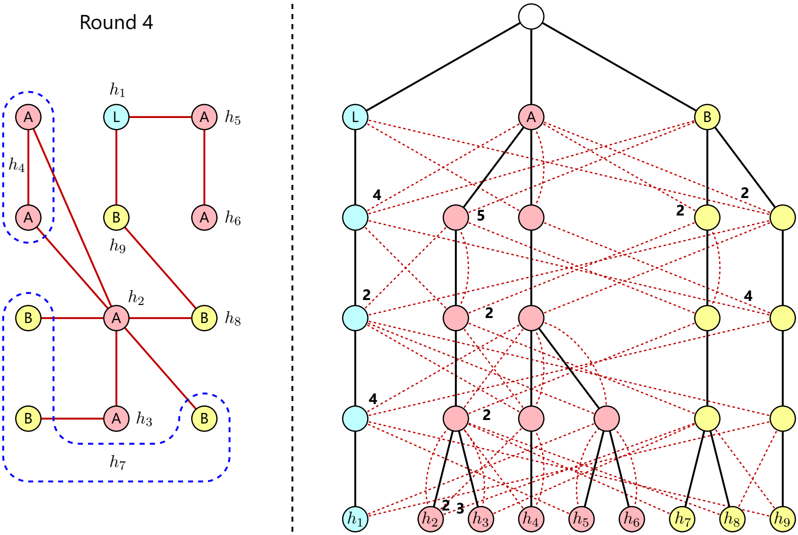
We introduced the *history tree* as a tool for studying dynamic networks.





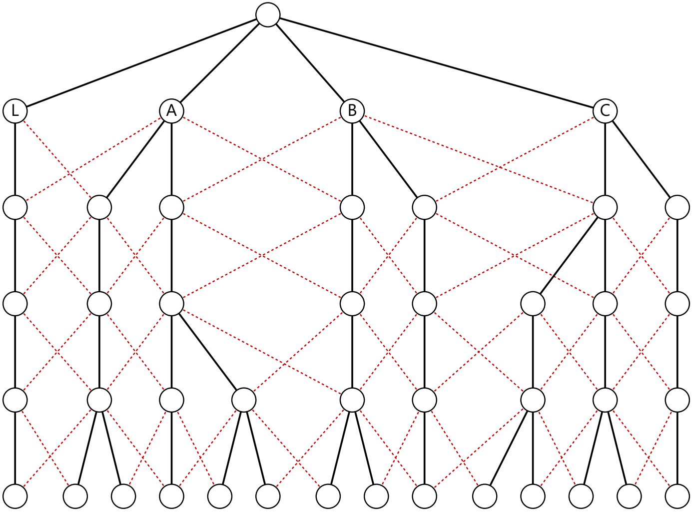
History tree

We introduced the *history tree* as a tool for studying dynamic networks.



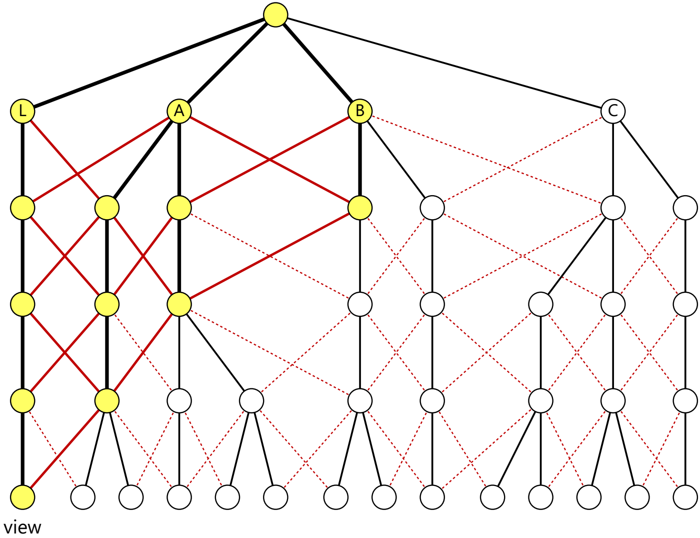
View of a history tree

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



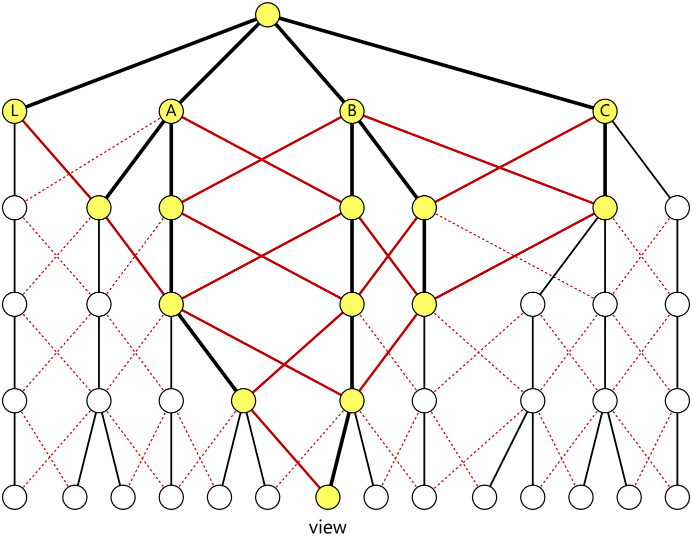
View of a history tree

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 $O(i^4)$.

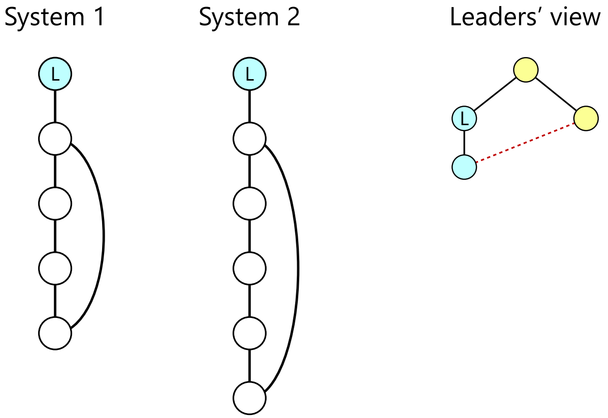
Observation

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.

Improved lower bound

Theorem

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



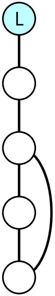
Round 1

Improved lower bound

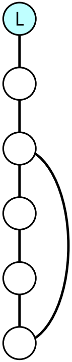
Theorem

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

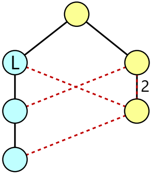
System 1



System 2



Leaders' view



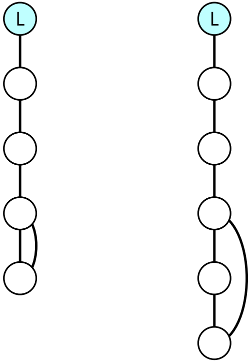
Round 2

Improved lower bound

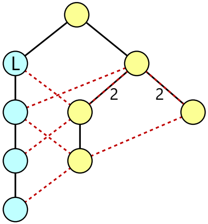
Theorem

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

System 1 System 2



Leaders' view



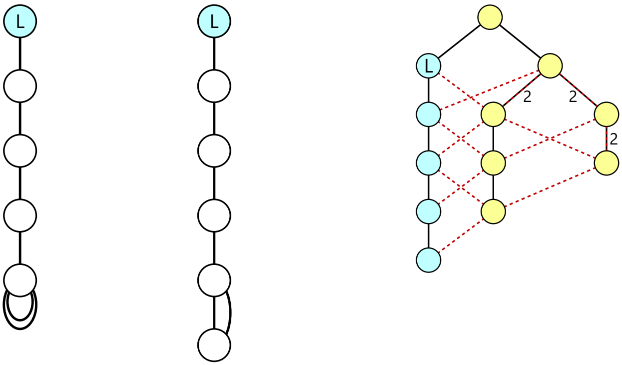
Round 3

Improved lower bound

Theorem

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

System 1 System 2 Leaders' view



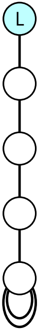
Round 4

Improved lower bound

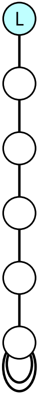
Theorem

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

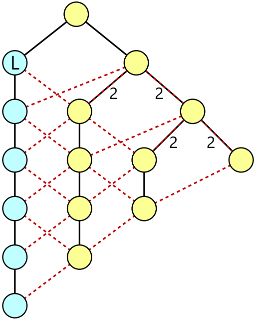
System 1



System 2



Leaders' view

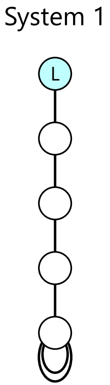


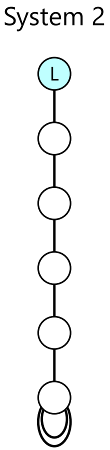
Round 5

Improved lower bound

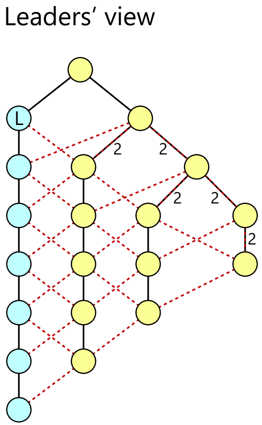
Theorem

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





Round 6

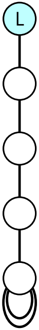


Improved lower bound

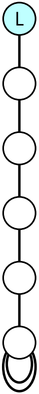
Theorem

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

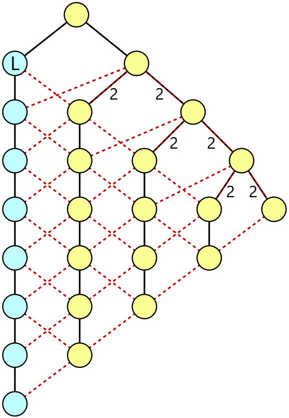
System 1



System 2



Leaders' view



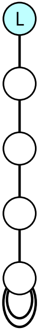
Round 7

Improved lower bound

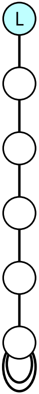
Theorem

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

System 1

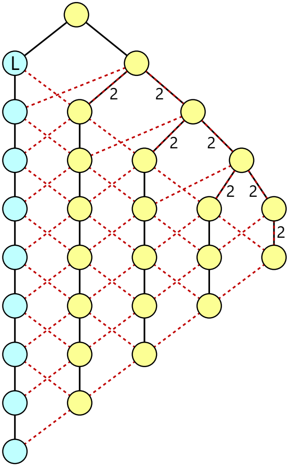


System 2



Round 8

Leaders' view



Counting algorithm: Overview

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

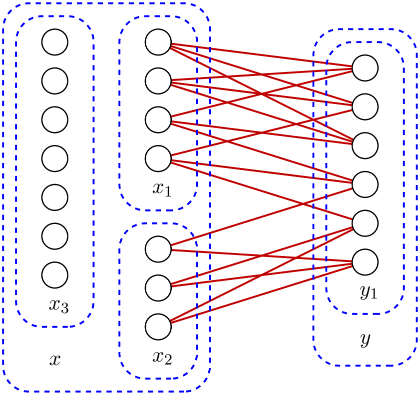
The algorithm is as follows:

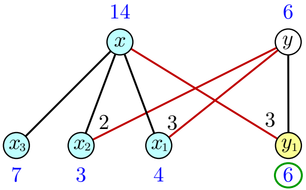
- Use the Leader's observations to make guesses on anonymities.
- In any set of n guesses, we can always identify a correct one.
- Once we have identified n correct guesses, we can use some of them to make new guesses on anonymities.
- Repeat until we have the anonymity of all visible branches of the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

The total running time is at most $3n$ rounds.

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.



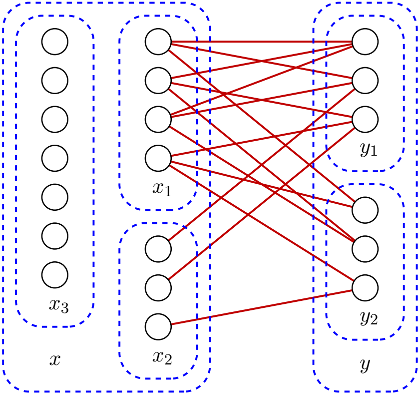


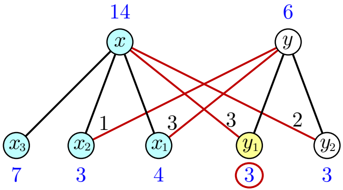
Guess on y_1: $\dfrac{4 \cdot 3 + 3 \cdot 2}{3} = 6$

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.



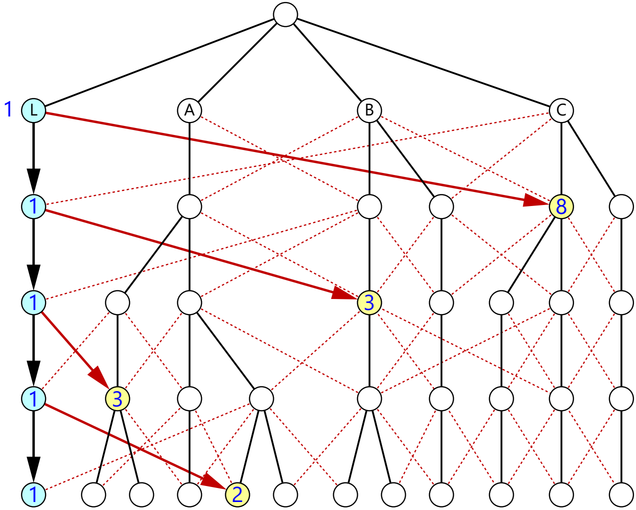


Guess on y_1: $\dfrac{4 \cdot 3 + 3 \cdot 1}{3} = 5$

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

Guessing anonymities from the Leader

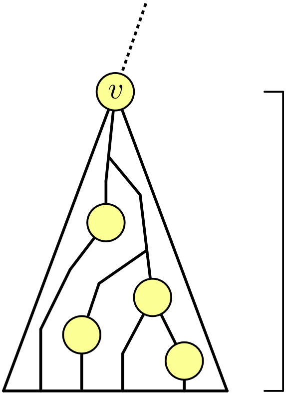
We can make one guess per round using the Leader's observations.



How do we know which guesses are correct?

Weight of a node

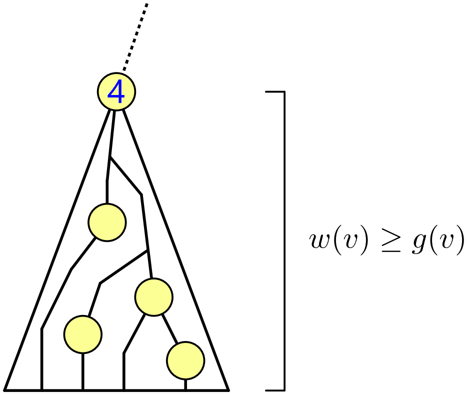
When a node v has a guess, we define its *weight* $w(v)$ as the number of nodes in the subtree hanging from v that have guesses.



$$w(v) = 5$$

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, $a(v) \geq \sum_i a(v_i)$.

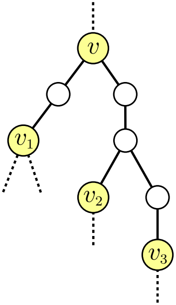
By the inductive hypothesis, $w(v_i) \leq a(v_i)$ for all i.

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

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. $\square$



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 $a(v) \leq g(v)$.

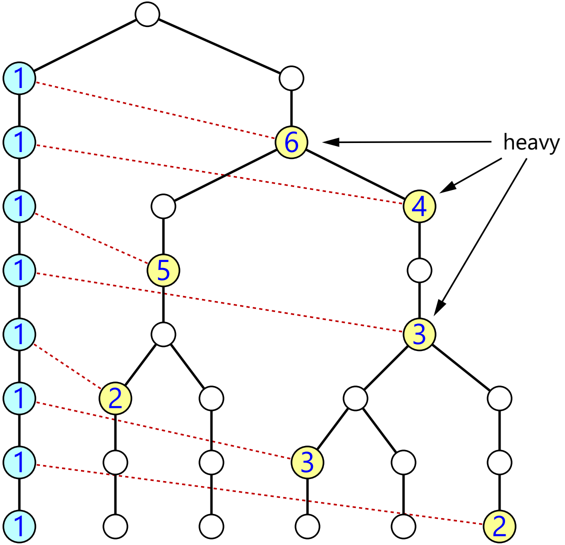
$g(v) \leq w(v) \leq a(v) \leq g(v)$, hence $g(v) = a(v)$. □

This corollary gives agents an algorithm 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 $n - 1$ guesses in total, i.e., by round n.

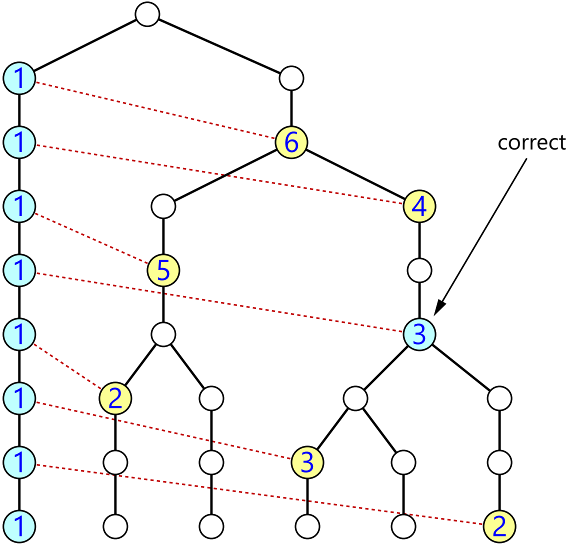
Criterion of correctness: Example

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



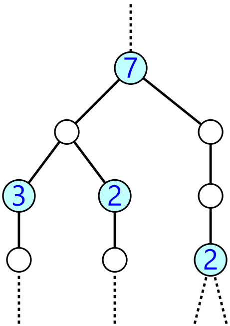
Criterion of correctness: Example

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



Propagation of guesses

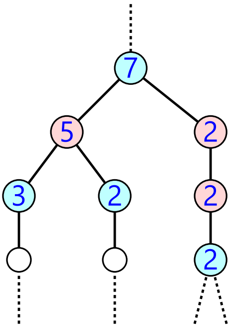
An *island* is a connected component of (a view of) the history tree that contains no leaves and does not contain the root.



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

Propagation of guesses

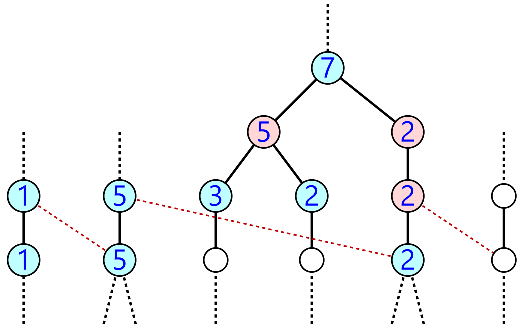
An *island* is a connected component of (a view of) the history tree that contains no leaves and does not contain the root.



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

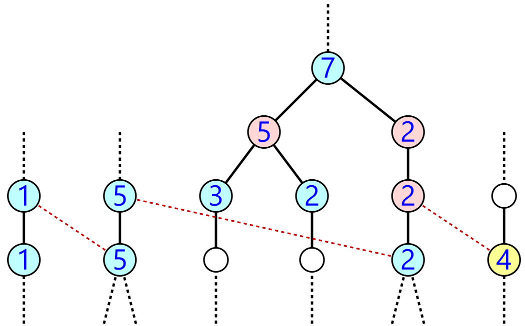
An *island* is a connected component of (a view of) the history tree that contains no leaves and does not contain the root.



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

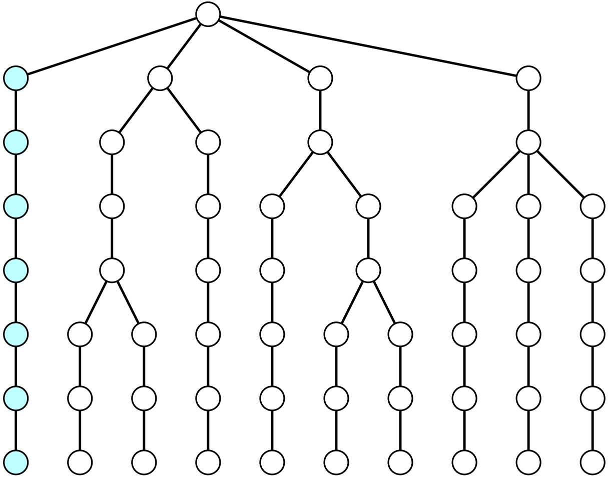
Propagation of guesses

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



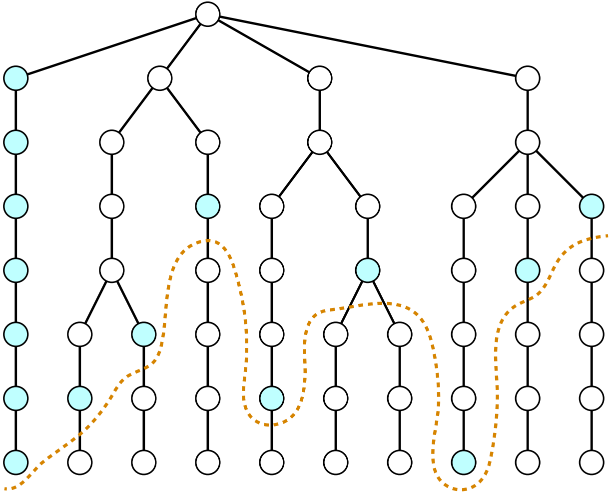
Since the network is connected at every round, we can make a new guess from one of the nodes in the island.

Propagation of guesses



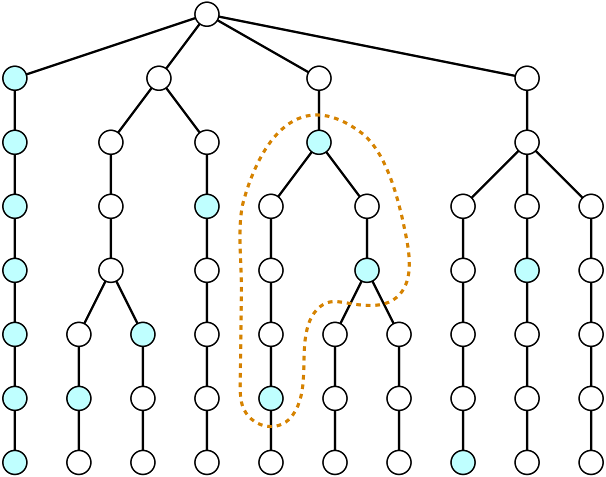
Suppose that there are $n-1$ nodes with necessarily correct guesses (other than the Leader ones). There are two cases:

Propagation of guesses



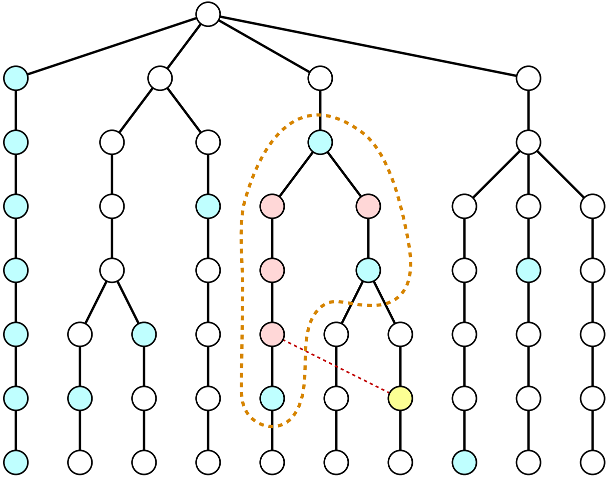
Either these nodes determine a *cut* of the history tree, in which case we have an estimate n' on n, given by their sum...

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 $n-1$ 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.

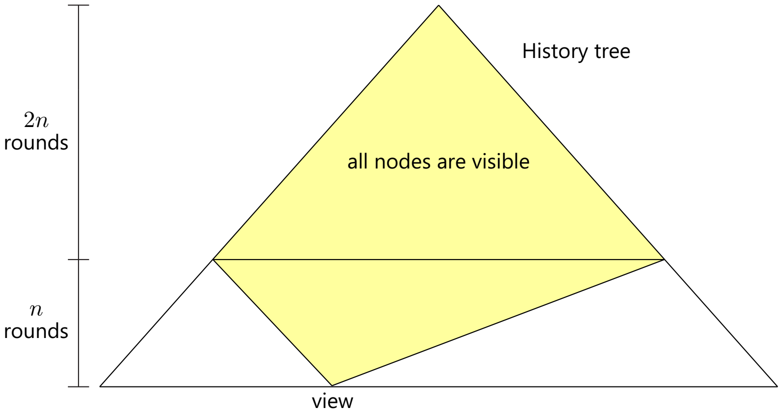
Once the second buffer is full, every new correct guess (blue node) either creates a new island or splits an island in two; in both cases, we can make a new guess (yellow node).

Therefore, within $2n$ 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.

Conclusions

Theorem

The Counting Problem cannot be solved in less than $2n-3$ rounds in a connected anonymous dynamic network with a Leader (with or without termination).

Theorem

Any problem that is solvable in a connected anonymous dynamic network with a Leader can be solved:

- *in $2n-2$ rounds without termination,*
- *in $3n$ rounds with termination.*

Additionally, internal states and messages have size $O(n^4)$.

Open Problem: Can we reduce the last $3n$ to $2n$?

Open Problem: What if the Leader is not unique, but there are ℓ indistinguishable Leaders (where ℓ is known by all agents)?