Distributed Computing
14 - LOCAL Variants

Mikaël Rabie
Université Paris Cité, IRIF

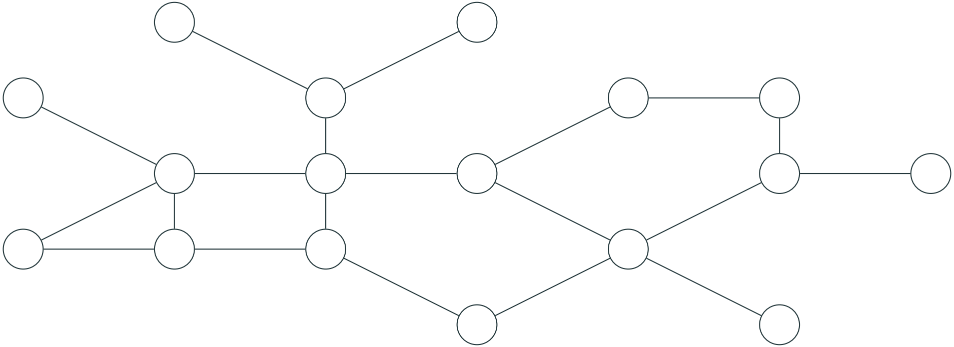
INSTITUT
DE RECHERCHE
EN INFORMATIQUE
FONDAMENTALE

Université de Paris

Sequential Complexity

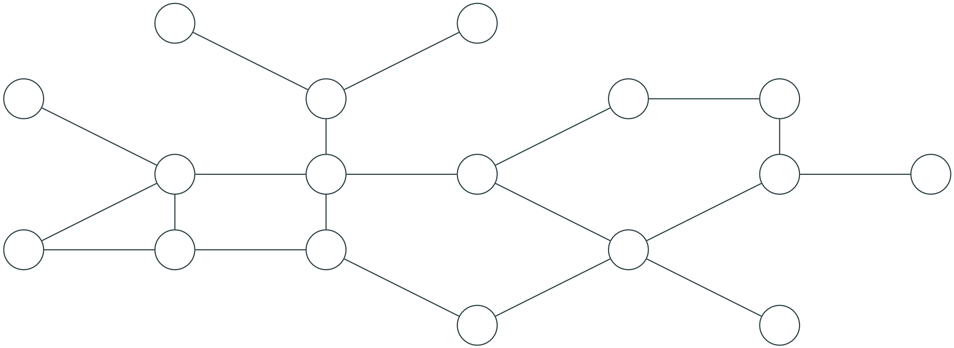
SLOCAL Model

- Each node is activated one after another, to compute its own output
- A node has access to the outputs already computed to produce its own
- Complexity : maximal radius needed among nodes



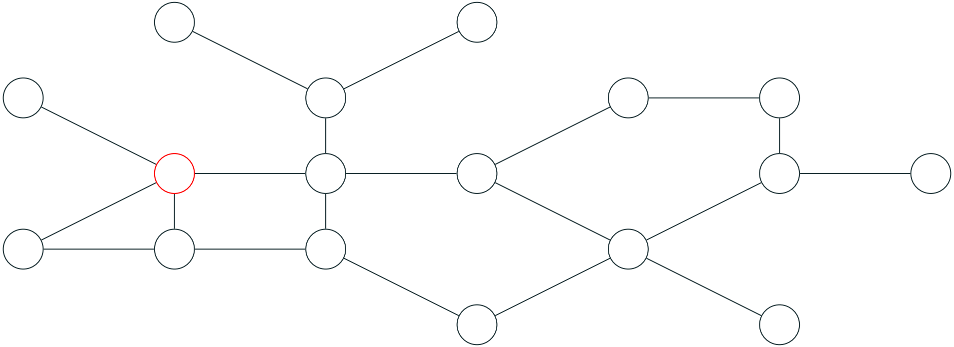
SLOCAL Model

- Each node is activated one after another, to compute its own output
- A node has access to the outputs already computed to produce its own
- Complexity : maximal radius needed among nodes
- **Greedy** problems can be solved in radius $O(1)$



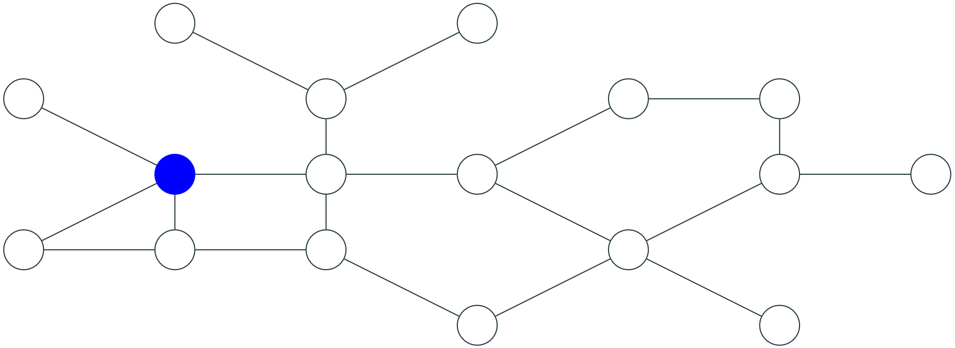
SLOCAL Model

- Each node is activated one after another, to compute its own output
- A node has access to the outputs already computed to produce its own
- Complexity : maximal radius needed among nodes
- **Greedy** problems can be solved in radius $O(1)$



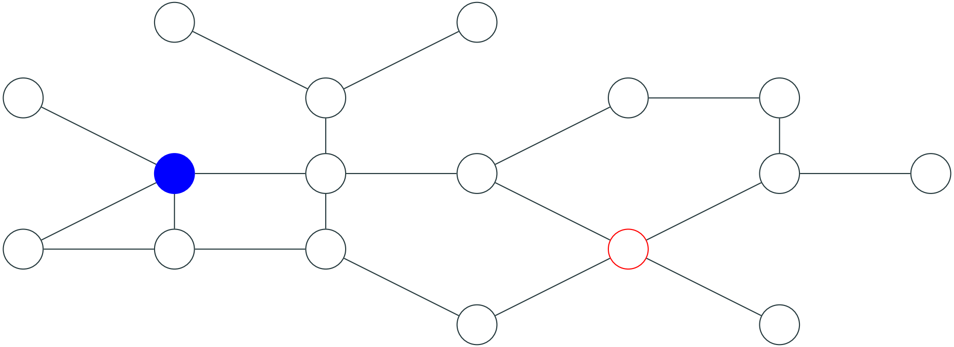
SLOCAL Model

- Each node is activated one after another, to compute its own output
- A node has access to the outputs already computed to produce its own
- Complexity : maximal radius needed among nodes
- **Greedy** problems can be solved in radius $O(1)$



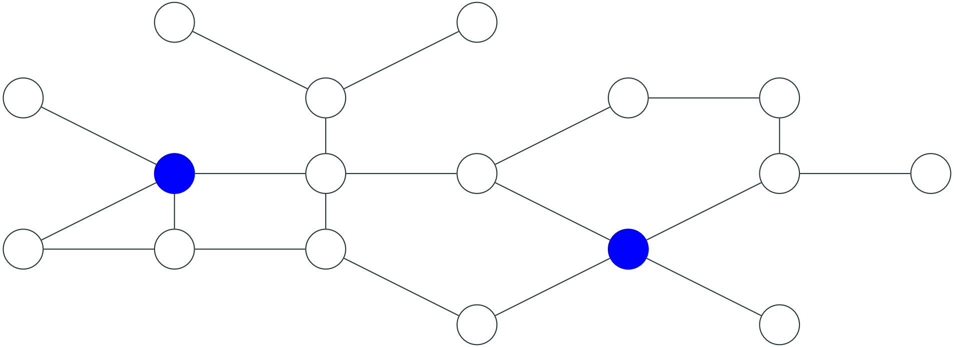
SLOCAL Model

- Each node is activated one after another, to compute its own output
- A node has access to the outputs already computed to produce its own
- Complexity : maximal radius needed among nodes
- **Greedy** problems can be solved in radius $O(1)$



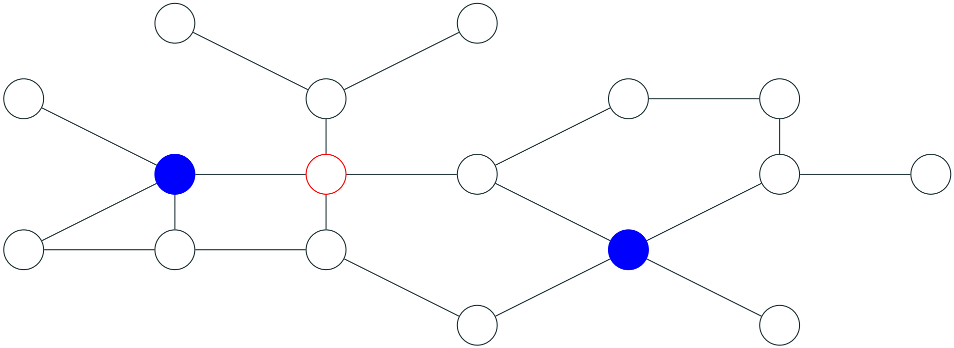
SLOCAL Model

- Each node is activated one after another, to compute its own output
- A node has access to the outputs already computed to produce its own
- Complexity : maximal radius needed among nodes
- **Greedy** problems can be solved in radius $O(1)$



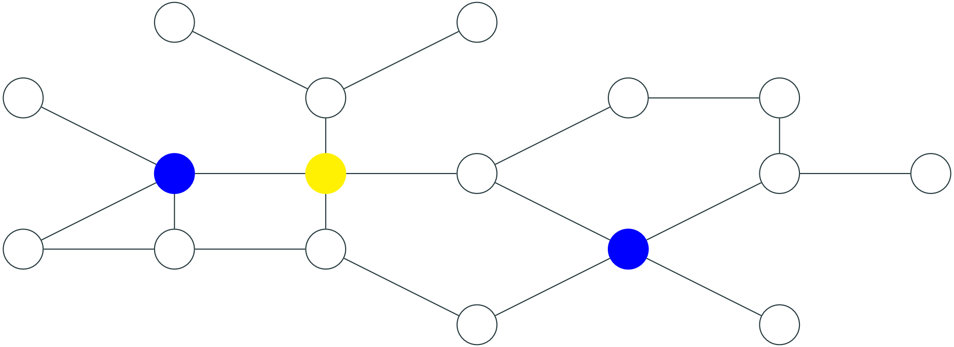
SLOCAL Model

- Each node is activated one after another, to compute its own output
- A node has access to the outputs already computed to produce its own
- Complexity : maximal radius needed among nodes
- **Greedy** problems can be solved in radius $O(1)$



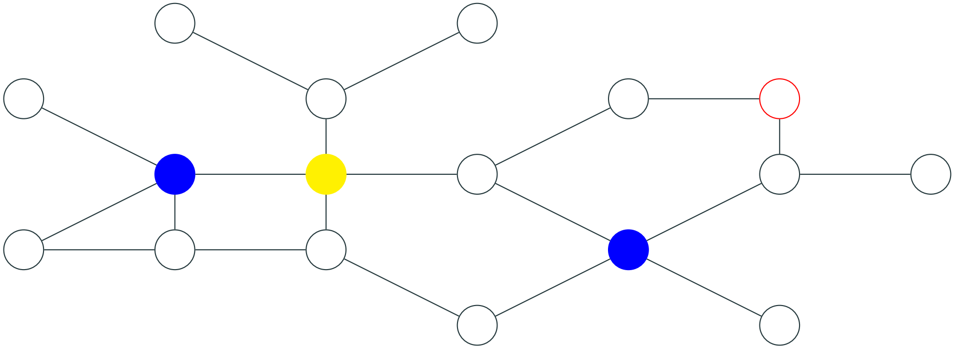
SLOCAL Model

- Each node is activated one after another, to compute its own output
- A node has access to the outputs already computed to produce its own
- Complexity : maximal radius needed among nodes
- **Greedy** problems can be solved in radius $O(1)$



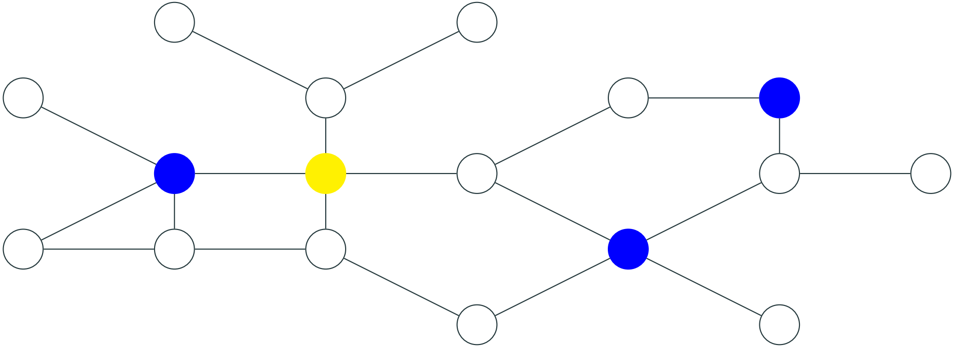
SLOCAL Model

- Each node is activated one after another, to compute its own output
- A node has access to the outputs already computed to produce its own
- Complexity : maximal radius needed among nodes
- **Greedy** problems can be solved in radius $O(1)$



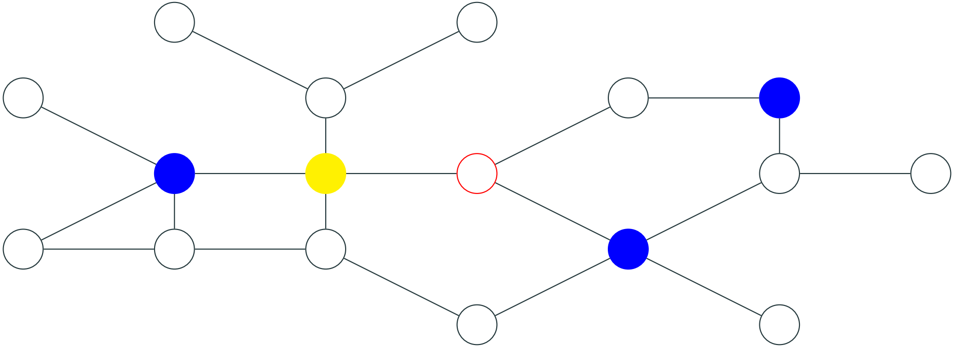
SLOCAL Model

- Each node is activated one after another, to compute its own output
- A node has access to the outputs already computed to produce its own
- Complexity : maximal radius needed among nodes
- **Greedy** problems can be solved in radius $O(1)$



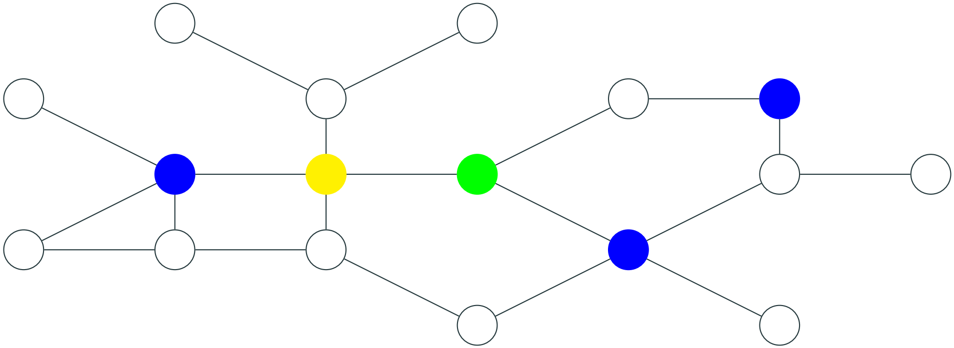
SLOCAL Model

- Each node is activated one after another, to compute its own output
- A node has access to the outputs already computed to produce its own
- Complexity : maximal radius needed among nodes
- **Greedy** problems can be solved in radius $O(1)$



SLOCAL Model

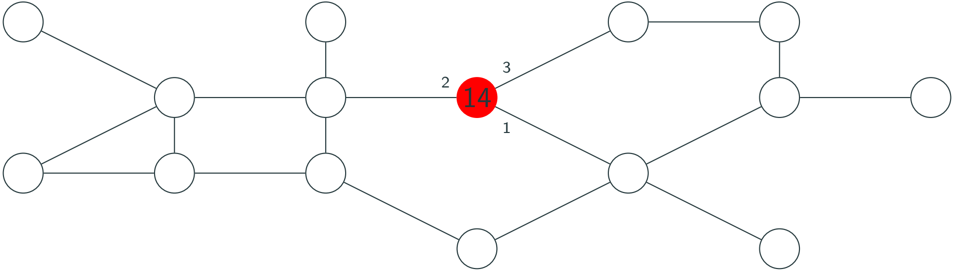
- Each node is activated one after another, to compute its own output
- A node has access to the outputs already computed to produce its own
- Complexity : maximal radius needed among nodes
- **Greedy** problems can be solved in radius $O(1)$



Volume Complexity

CentLOCAL Model

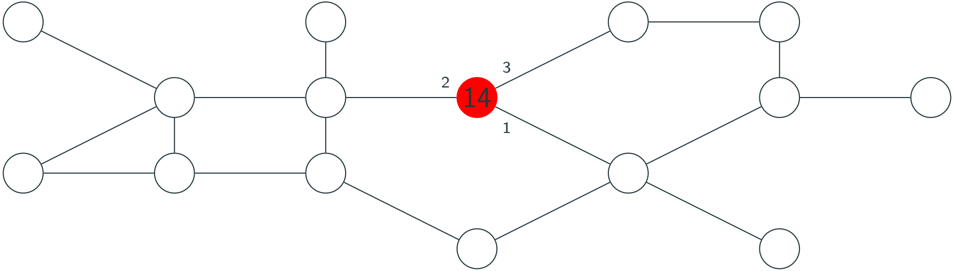
- In parallel, each node v :
  - Knows its own Id_v and degree d_{Id_v}
  - At each step, they send a request (Id_u, k), with $k \leq d_{Id_u}$
  - They get (Id_w, d_{Id_w}, k') such that $(u, v) \in E$ are connected by port k from u and k' from w
- Complexity : maximal number of requests from a node



CentLOCAL Model

- In parallel, each node v :
  - Knows its own Id_v and degree d_{Id_v}
  - At each step, they send a request (Id_u, k), with $k \leq d_{Id_u}$
  - They get (Id_w, d_{Id_w}, k') such that $(u, v) \in E$ are connected by port k from u and k' from w
- Complexity : maximal number of requests from a node

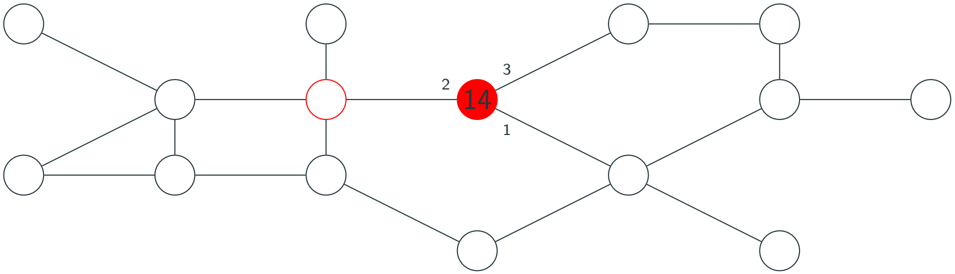
Request : (14,2)



CentLOCAL Model

- In parallel, each node v :
  - Knows its own Id_v and degree d_{Id_v}
  - At each step, they send a request (Id_u, k), with $k \leq d_{Id_u}$
  - They get (Id_w, d_{Id_w}, k') such that $(u, v) \in E$ are connected by port k from u and k' from w
- Complexity : maximal number of requests from a node

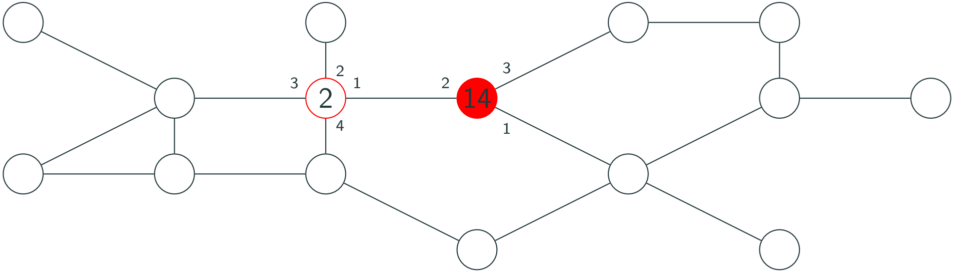
Request : (14,2)



CentLOCAL Model

- In parallel, each node v :
  - Knows its own Id_v and degree d_{Id_v}
  - At each step, they send a request (Id_u, k), with $k \leq d_{Id_u}$
  - They get (Id_w, d_{Id_w}, k') such that $(u, v) \in E$ are connected by port k from u and k' from w
- Complexity : maximal number of requests from a node

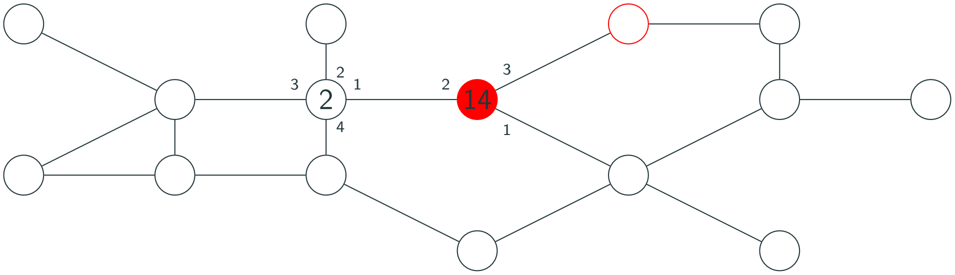
Request : (14,2) $\Rightarrow$ (2,4,1)



CentLOCAL Model

- In parallel, each node v :
  - Knows its own Id_v and degree d_{Id_v}
  - At each step, they send a request (Id_u, k), with $k \leq d_{Id_u}$
  - They get (Id_w, d_{Id_w}, k') such that $(u, v) \in E$ are connected by port k from u and k' from w
- Complexity : maximal number of requests from a node

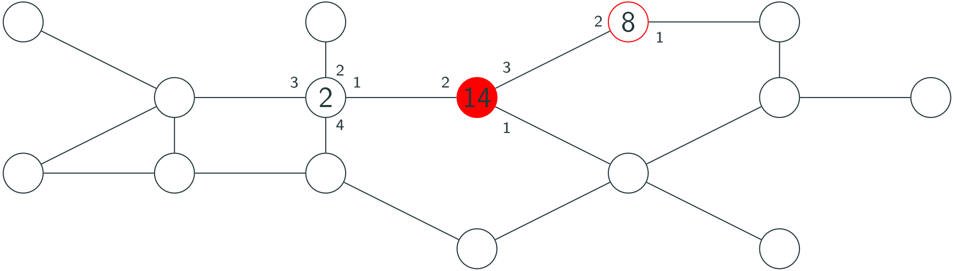
Request : (14,3)



CentLOCAL Model

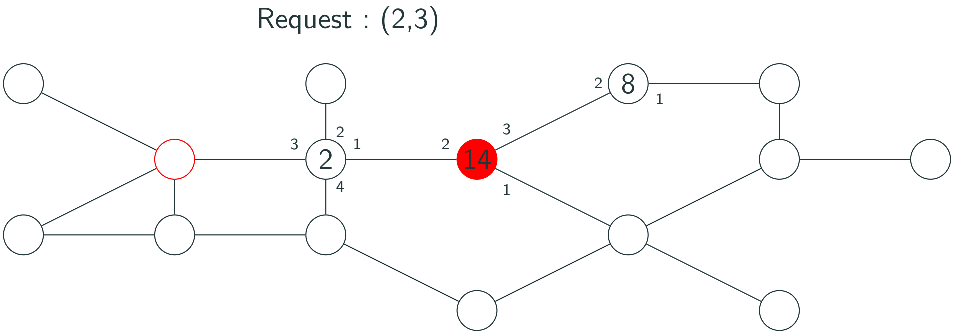
- In parallel, each node v :
  - Knows its own Id_v and degree d_{Id_v}
  - At each step, they send a request (Id_u, k), with $k \leq d_{Id_u}$
  - They get (Id_w, d_{Id_w}, k') such that $(u, v) \in E$ are connected by port k from u and k' from w
- Complexity : maximal number of requests from a node

Request : (14,3) $\Rightarrow$ (8,2,2)



CentLOCAL Model

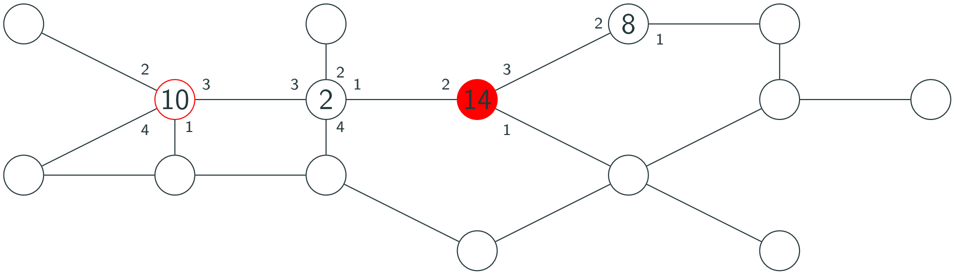
- In parallel, each node v :
  - Knows its own Id_v and degree d_{Id_v}
  - At each step, they send a request (Id_u, k), with $k \leq d_{Id_u}$
  - They get (Id_w, d_{Id_w}, k') such that $(u, v) \in E$ are connected by port k from u and k' from w
- Complexity : maximal number of requests from a node



CentLOCAL Model

- In parallel, each node v :
  - Knows its own Id_v and degree d_{Id_v}
  - At each step, they send a request (Id_u, k), with $k \leq d_{Id_u}$
  - They get (Id_w, d_{Id_w}, k') such that $(u, v) \in E$ are connected by port k from u and k' from w
- Complexity : maximal number of requests from a node

Request : (2,3) $\Rightarrow$ (10,4,3)



Greedy Problems

Problem A can be solved in time $\Theta(f(n))$ in the LOCAL model

$\Rightarrow$ A can be solved in time in the CentLOCAL model

Greedy Problems

Problem A can be solved in time $\Theta(f(n))$ in the LOCAL model

$\Rightarrow A$ can be solved in time $\Omega(f(n))$ and $O\left(\Delta^{f(n)}\right)$ in the CentLOCAL model

Greedy Problems

Problem A can be solved in time $\Theta(f(n))$ in the LOCAL model
$\Rightarrow$ A can be solved in time $\Omega(f(n))$ and $O\left(\Delta^{f(n)}\right)$ in the CentLOCAL model

Even *et. al* **(2018)**

There is a CentLOCAL algorithm in time $O(\Delta \times log^* n + \Delta^3)$ for $\leq \Delta^2$-coloring a graph.
There is a CentLOCAL algorithm in time $O(\Delta \times log^* n + \Delta^3)$ for orienting a graph where the longer oriented path is of length $\leq \Delta^2$.
Any greedy problem can be solved in time $O(f(\Delta) \times log^* n)$.

Complexity Gap

Rosenbaum and Suomela (2020)
In the CentLOCAL model, if n is not given in advance and identifiers do not require to be polynomial in n, there is no problem whose time complexity is in $\omega(log^* n) \cap o(n)$.

Complexity Gap

Rosenbaum and Suomela (2020)

In the CentLOCAL model, if n is not given in advance and identifiers do not require to be polynomial in n, there is no problem whose time complexity is in $\omega(log^* n) \cap o(n)$.

- Take N such that $T(N) \ll N$
- Do a distance N-coloring
- Simulate the algorithm with the new identifiers

Mendability

Mendable Problems

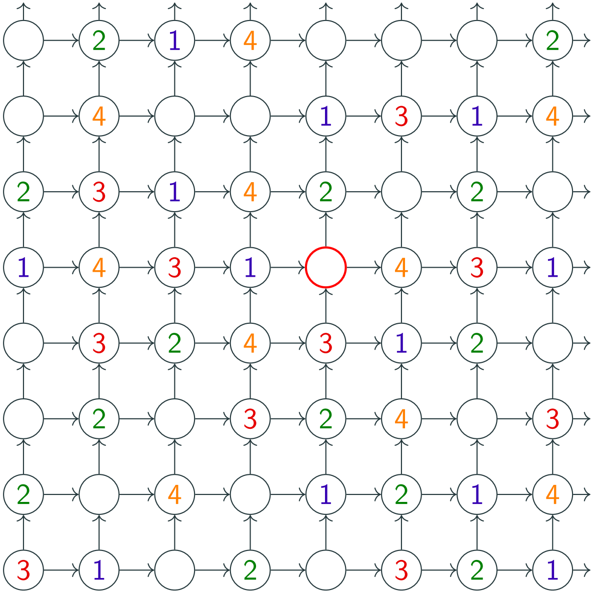
$\Gamma^* : V \to \mathcal{O} \cup \{\perp\}$ is a **Partial Solution** if :

- $\mathcal{O}$ is the Output Set,
- $\forall u \in V : \Gamma^*(u) \neq \perp \Rightarrow$ we can complete the labels of the neighbors of u.

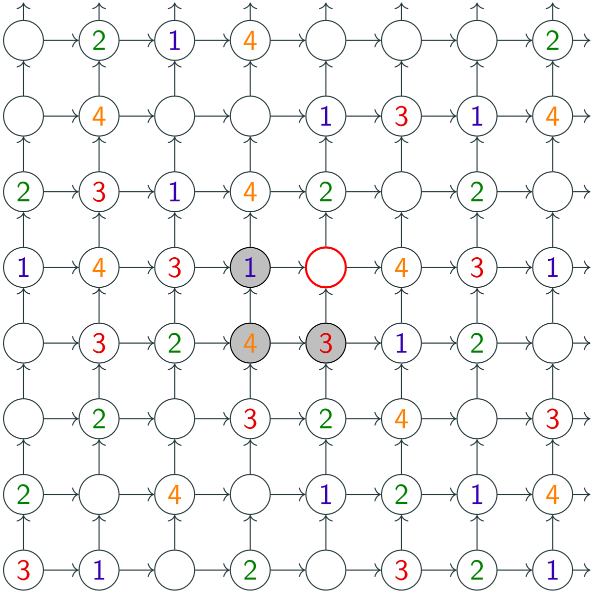
A problem is T-**Mendable** if, from any partial solution Γ^* and any $v \in V$ such that $\Gamma^*(v) = \perp$, there exists Γ' :

- $\Gamma'(v) \neq \perp$
- $\forall u \neq v$, $\Gamma'(u) = \perp \Leftrightarrow \Gamma^*(u) = \perp$
- $\forall u \in V$, $dist(u, v) > T \Rightarrow \Gamma'(u) = \Gamma^*(u)$

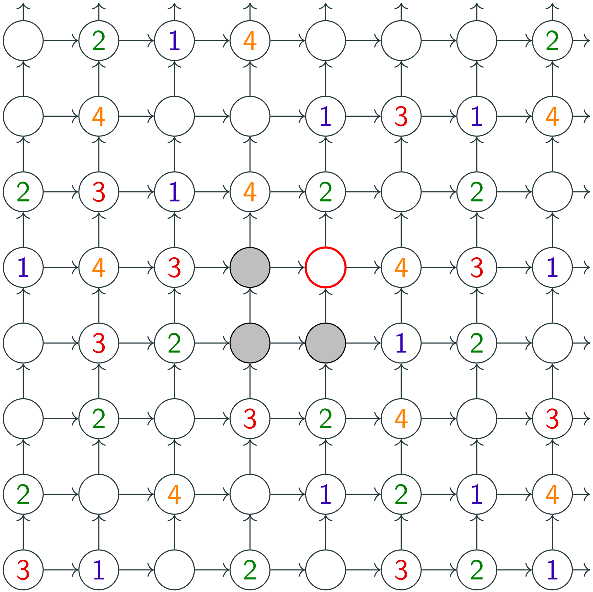
4-coloring the Grid



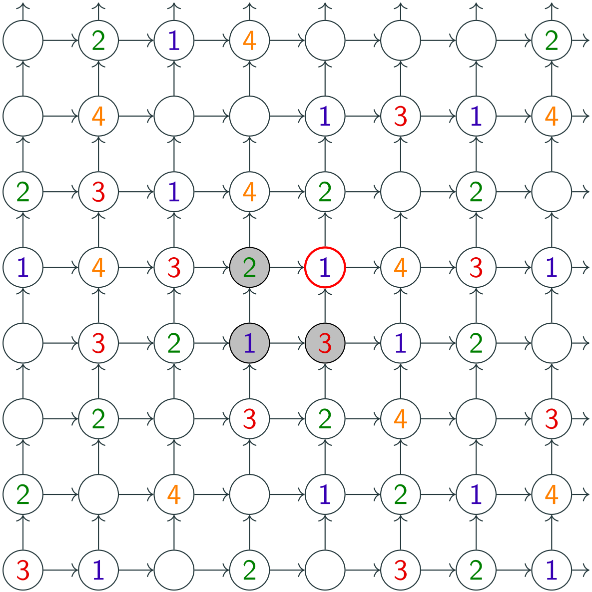
4-coloring the Grid



4-coloring the Grid



4-coloring the Grid



Mendable into LOCAL

Balliu *et. al* (2022)

Let Π be a T-mendable LCL problem. Π can be solved in rounds in the LOCAL model if we are given a distance- coloring.

Mendable into LOCAL

Balliu ***et. al*** **(2022)**

Let Π be a T-mendable LCL problem. Π can be solved in $O\left(T\Delta^{2T}\right)$ rounds in the LOCAL model if we are given a distance-$2T+1$ coloring.

Mendable into LOCAL

Balliu *et. al* (2022)
Let Π be a T-mendable LCL problem. Π can be solved in $O\left(T\Delta^{2T}\right)$ rounds in the LOCAL model if we are given a distance-$2T+1$ coloring.

Balliu *et. al* (2022)
Let Π be a $O(1)$-mendable LCL problem. Π can be solved in $O\left(\log^* n\right)$ rounds in the LOCAL model on bounded degree graphs.

From $\log^* n$ to Mendability

On paths and cycles, are all $O(\log^* n)$ problems mendable?

From $\log^* n$ to Mendability

On paths and cycles, are all $O(\log^* n)$ problems mendable ?

No : 3-color with $\{1, 2, 3\}$ or 2-color with $\{A, B\}$.

From $\log^* n$ to Mendability

On paths and cycles, are all $O(\log^* n)$ problems mendable ?

No : 3-color with $\{1, 2, 3\}$ or 2-color with $\{A, B\}$.

Balliu *et. al* **(2022)**
Suppose Π is an LCL problem on directed cycles with no input. If Π is $O(\log^* n)$-solvable, we can define a new LCL problem Π' with the same round complexity, such that a solution for Π' is also a solution for Π, and Π' is $O(1)$-mendable.

The Case of Trees

Balliu ***et. al*** **(2022)**
In trees, there are exactly three classes : $O(1)$-mendable, $\Theta(\log n)$-mendable, and $\Theta(n)$-mendable problems.

3-coloring the rooted tree is

The Case of Trees

Balliu *et. al* **(2022)**
In trees, there are exactly three classes : $O(1)$-mendable, $\Theta(\log n)$-mendable, and $\Theta(n)$-mendable problems.

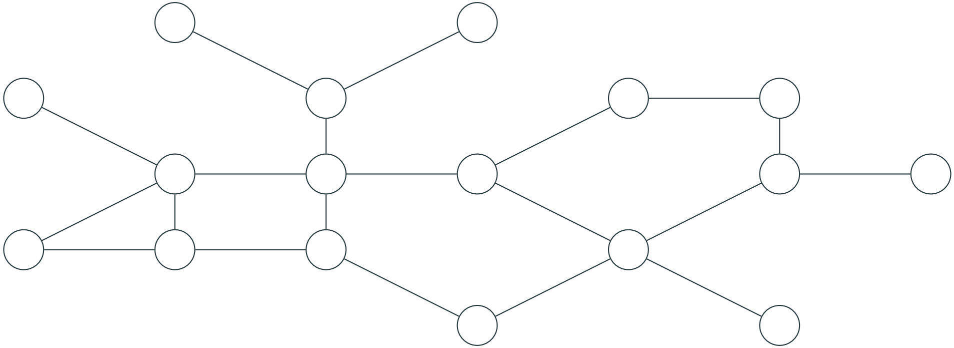
3-coloring the rooted tree is $O(n)$-mendable.
There exists a $O(1)$-mendable problem Π' that projects its solutions to a 3-coloring :

- A node is **monochromatic** if both its children have the same color.
- Otherwise, the node is **mixed**.
- Π' only accept connected components of mixed nodes of height $\leq k$.

Waking Up Complexity

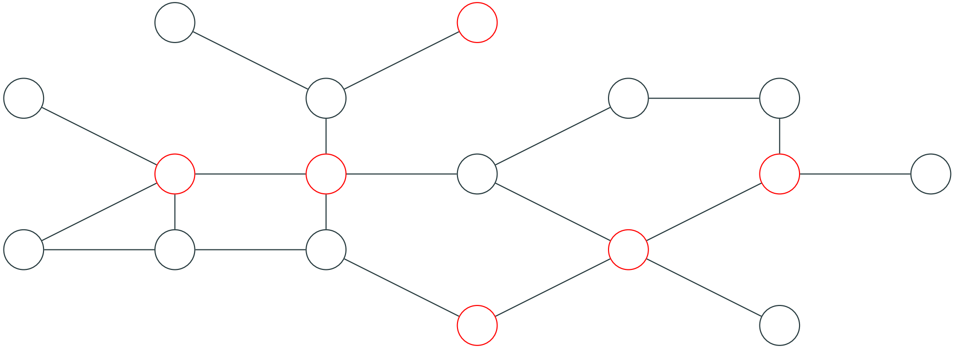
Sleeping LOCAL Model

- At each round, a node decides if it is active or not
- A communicates only with its active neighbors
- Complexity : maximal number of active rounds for a single node



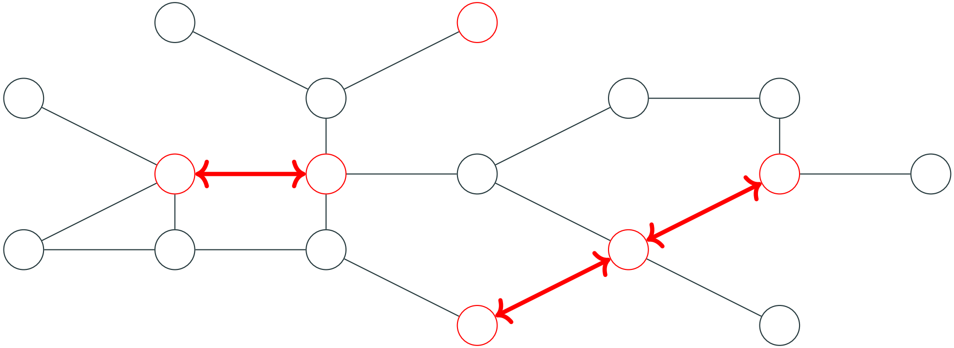
Sleeping LOCAL Model

- At each round, a node decides if it is active or not
- A communicates only with its active neighbors
- Complexity : maximal number of active rounds for a single node



Sleeping LOCAL Model

- At each round, a node decides if it is active or not
- A communicates only with its active neighbors
- Complexity : maximal number of active rounds for a single node



A Link with SLOCAL

$\Delta + 1$-coloring can be solved in $O(\Delta)$ rounds

A Link with SLOCAL

$\Delta + 1$-coloring can be solved in $O(\Delta)$ rounds :

- Round 1 : all nodes are activated. Know their identifiers and their neighbours'.
- Node of Identifier Id wakes up at round $Id + 1$ to know their neighbours' colors.
- Neighbours of node of identifier Id also wakes up at that round.

Drawback : The round complexity is $O(M)$, M being the maximal identifier.

A Link with SLOCAL

$\Delta + 1$-coloring can be solved in $O(\Delta)$ rounds :

- Round 1 : all nodes are activated. Know their identifiers and their neighbours'.
- Node of Identifier Id wakes up at round $Id + 1$ to know their neighbours' colors.
- Neighbours of node of identifier Id also wakes up at that round.

Drawback : The round complexity is $O(M)$, M being the maximal identifier.

Problem A can be solved in time $f(n)$ in the SLOCAL model
$\Rightarrow$ A can be solved in time in the Sleeping LOCAL model.

A Link with SLOCAL

$\Delta + 1$-coloring can be solved in $O(\Delta)$ rounds :

- Round 1 : all nodes are activated. Know their identifiers and their neighbours'.
- Node of Identifier Id wakes up at round $Id + 1$ to know their neighbours' colors.
- Neighbours of node of identifier Id also wakes up at that round.

Drawback : The round complexity is $O(M)$, M being the maximal identifier.

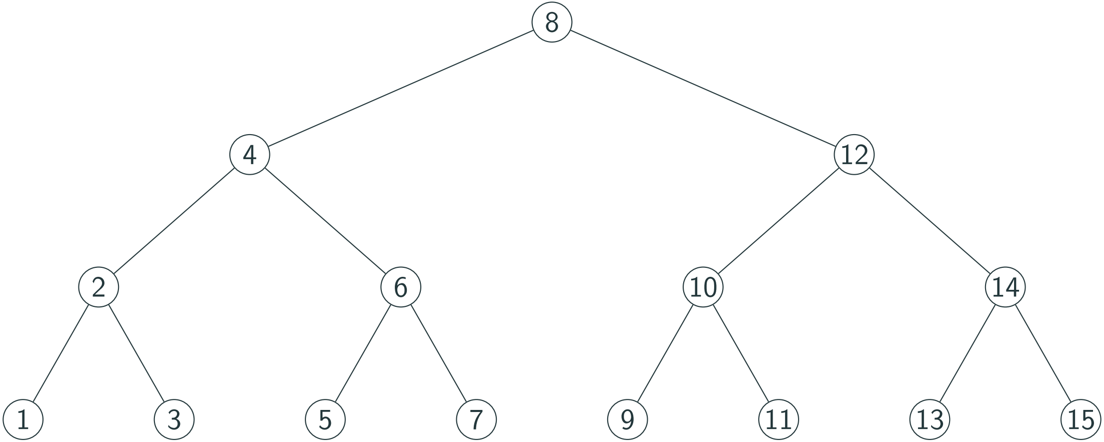
Problem A can be solved in time $f(n)$ in the SLOCAL model
$\Rightarrow A$ can be solved in time $O\left(f(n)\Delta^{f(n)}\right)$ in the Sleeping LOCAL model.

$\log \Delta$-coloring

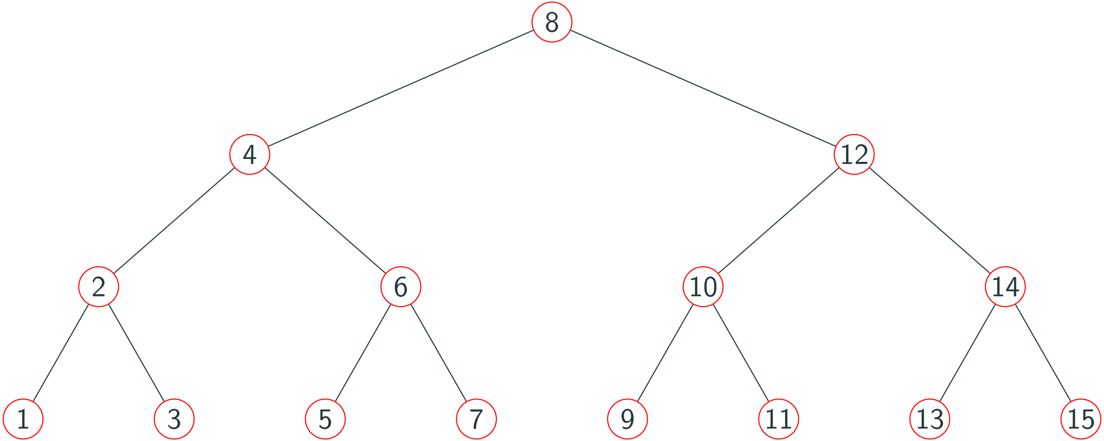
Barenboim and Maimon (2021)

Given a Δ^k-coloring of the graph, we can compute a $(\Delta + 1)$-coloring in $O(\log \Delta)$ awaken rounds and $O(\Delta^k)$ rounds in the Sleeping LOCAL model.

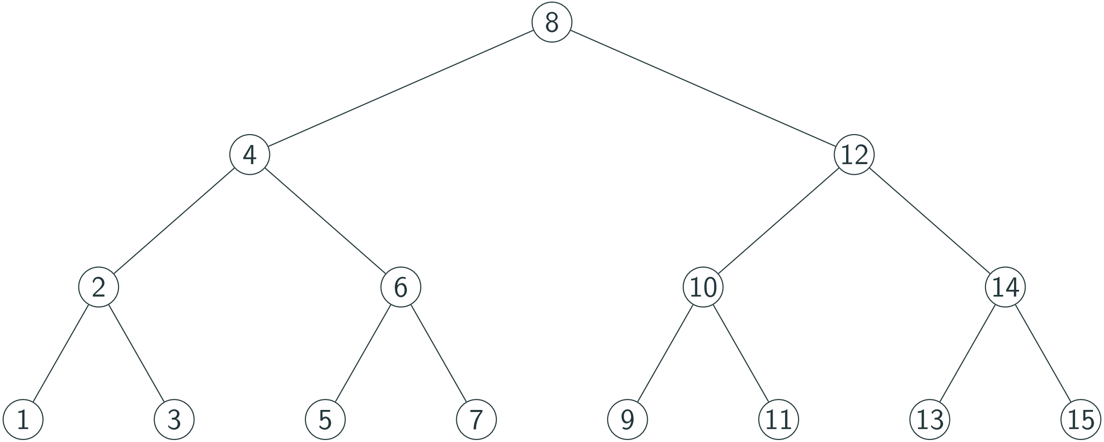
log Δ-coloring



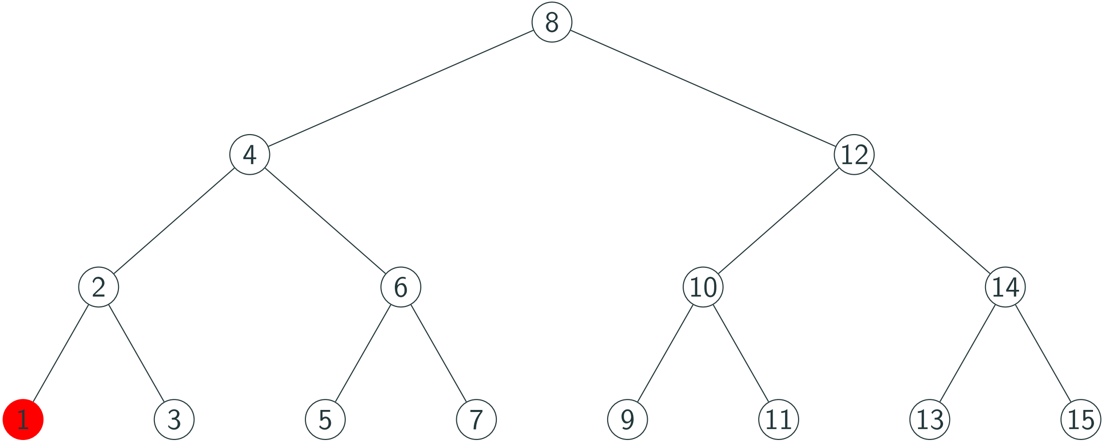
log Δ-coloring



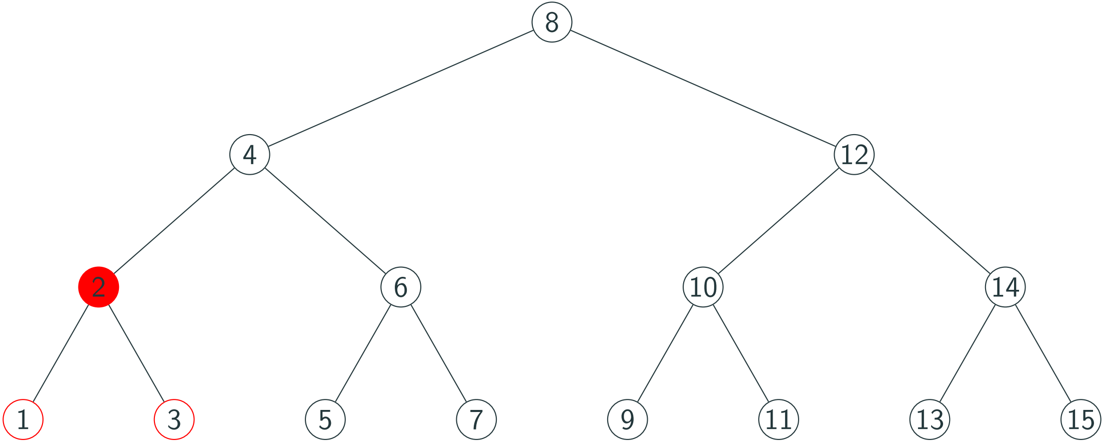
log Δ-coloring

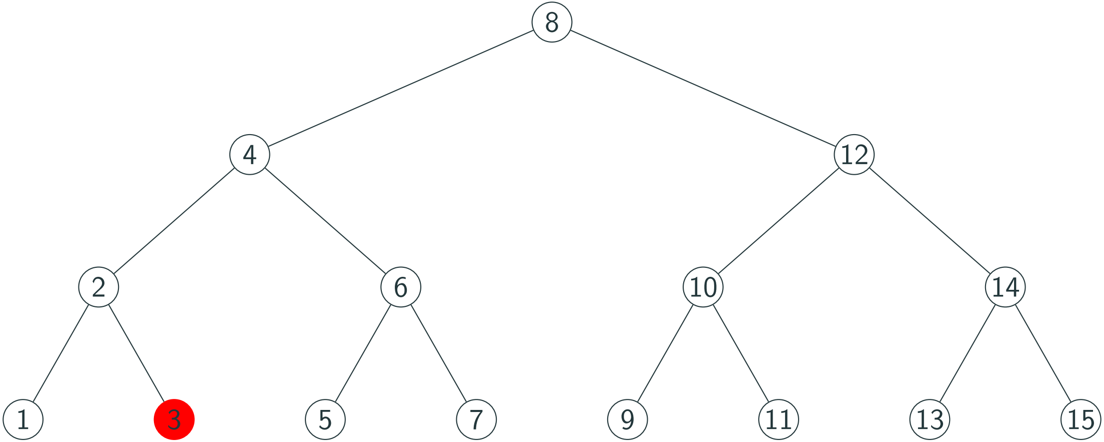


log Δ-coloring

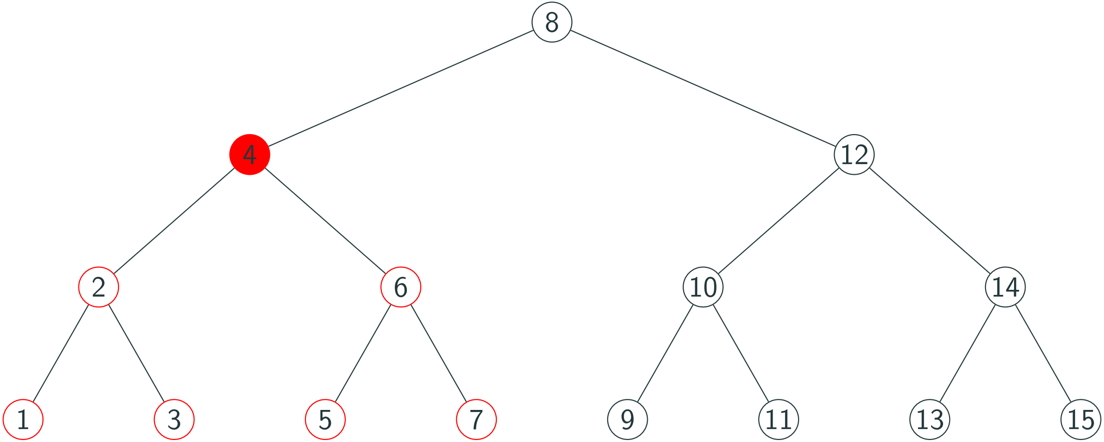


log Δ-coloring




8
4
12
2
6
10
14
1
3
5
7
9
11
13
15

log Δ-coloring



Full Knowledge of the Graph

Barenboim and Maimon (2021)
Any graph problem can be solved in $O(\log n)$ rounds in the Sleeping LOCAL model.

Full Knowledge of the Graph

Barenboim and Maimon (2021)
Any graph problem can be solved in $O(\log n)$ rounds in the Sleeping LOCAL model.

Distributed Layered Tree (DLT) - Oriented Spanning Tree such as :

- Each vertex has a label
- The label of a vertex is bigger than its parent's
- Each vertex knows the label of its neighbours in the tree

Full Knowledge of the Graph

Barenboim and Maimon (2021)
Any graph problem can be solved in $O(\log n)$ rounds in the Sleeping LOCAL model.

Distributed Layered Tree (DLT) - Oriented Spanning Tree such as :

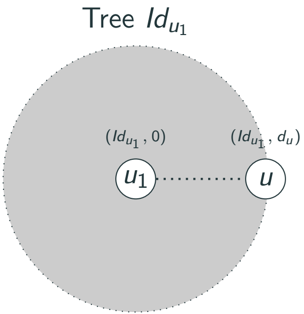
- Each vertex has a label
- The label of a vertex is bigger than its parent's
- Each vertex knows the label of its neighbours in the tree

Constant Coordination
Broadcast and Convergecast can be done in $O(1)$ rounds in a DLT.

Building a DLT

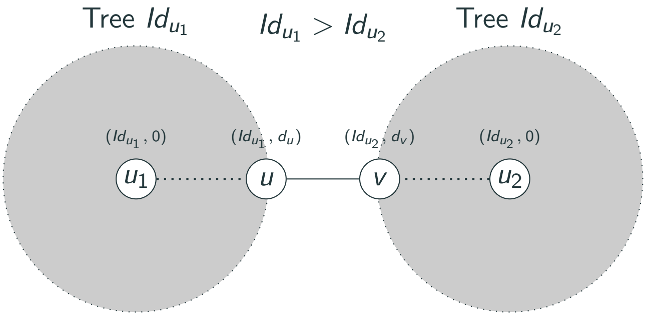
Barenboim and Maimon (2021)
A DLT can be built in $O(\log n)$ awaken rounds in the Sleeping LOCAL model.

Building a DLT



- Labels are of the form (a, b), ordered lexicographically.
- At the beginning, all nodes have label $(Id(u), 0)$.
- At the beginning of each expand step, all nodes of a subtree T are of the form $(L(T), b)$.

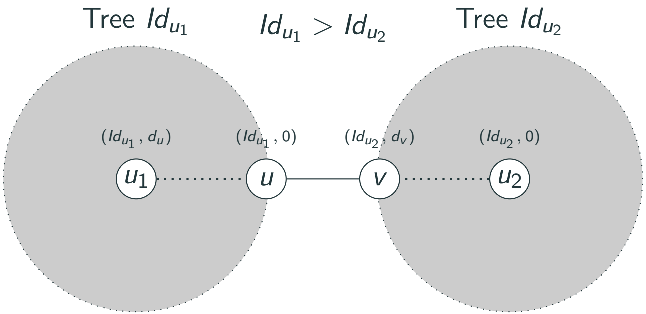
Building a DLT



- Repeat $\log n$ times :

1. Select a neighbour Tree T' with smaller label ($Id_{u_1} > Id_{u2}$).

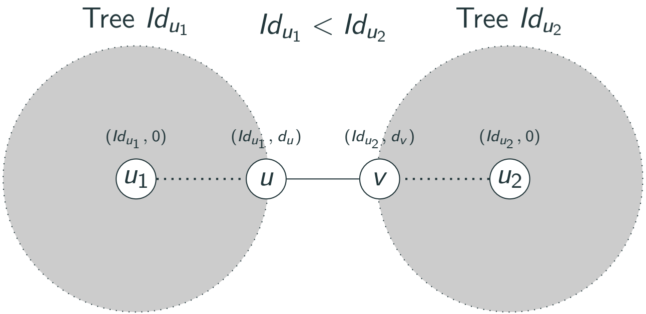
Building a DLT



- Repeat $\log n$ times :

2. Merge T and T', using an edge (u, v).

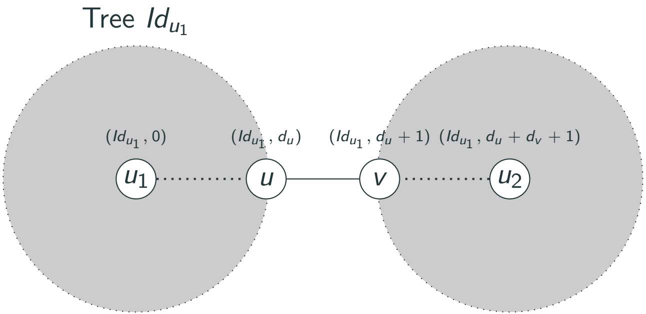
Building a DLT



- Repeat $\log n$ times :

3. If T could not choose a neighbour and was not selected
   T chooses a tree T' to join using an edge (u, v).
   This forms a star of trees around $T' \Rightarrow O(1)$ merge rounds.

Building a DLT



- Repeat $\log n$ times :

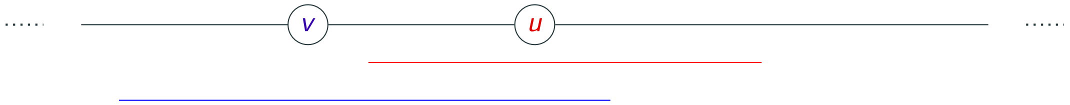
4. All nodes learn their new neighbours in the tree.
5. Convergecast to gather the new structure of the component C to the root r.
6. Broadcast a new labelling $(L(r), dist(r))$.

Sleeping Lower Bound

Augustine *et. al* (2022)
Any algorithm to solve 2-coloring with probability exceeding $1/8$ on a ring network requires $\Omega(\log n)$ awake time.

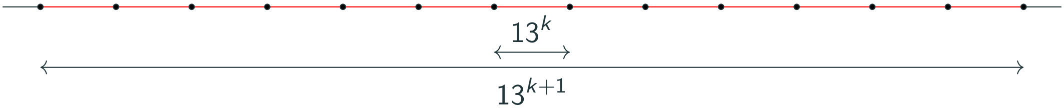
Sleeping Lower Bound

Augustine *et. al* (2022)
Any algorithm to solve 2-coloring with probability exceeding $1/8$ on a ring network requires $\Omega(\log n)$ awake time.



- After k rounds, a node knows about some segment that includes itself
- No node v on the left of u in the path can know more than u on its right

Sleeping Lower Bound



By induction : For any k, for any segment I of 13^k nodes, there exists, with probability $\mathcal{P} > 1/2$, a node $u \in I$ who knows less than I after k rounds.

Sleeping Lower Bound

By induction : For any k, for any segment I of 13^k nodes, there exists, with probability $\mathcal{P} > 1/2$, a node $u \in I$ who knows less than I after k rounds.

- Probability that it is true on 5 of the 13 subsegments is at least $5/6$
- Probability that B, C or D wakes up before A and E is at least $1/2$

Sleeping Lower Bound

By induction : For any k, for any segment I of 13^k nodes, there exists, with probability $\mathcal{P} > 1/2$, a node $u \in I$ who knows less than I after k rounds.

- Probability that it is true on 5 of the 13 subsegments is at least $5/6$
- Probability that B, C or D wakes up before A and E is at least $1/2$

Question : What are the compexity classes on paths and rings ?

Trade-Off

Find the possible trade-off between awaken and usual rounds to resolve a problem.

$(\Delta + 1)$-coloring :

Awaken rounds	Rounds
$O(\Delta)$	$O(M)$
$O(\log n)$	$\Omega(M)$
$O(\log^* n + \log \Delta)$	$O(\log^* n + \text{poly}\, \Delta)$

Trade-Off

Find the possible trade-off between awaken and usual rounds to resolve a problem.

$(\Delta + 1)$-coloring :

Awaken rounds	Rounds
$O(\Delta)$	$O(M)$
$O(\log n)$	$\Omega(M)$
$O(\log^* n + \log \Delta)$	$O(\log^* n + \text{poly } \Delta)$

Dufoulon, Moses, Pandurangan (2023)

Maximal Independent Set :

	Sleeping-Rand-MIS-1	Sleeping-Rand-MIS-2
Node-averaged awake complexity	$O(1)$	$O(1)$
Worst-case awake complexity	$O(\log \log n)$	$O((\log \log n) \log^* n)$
Total round complexity	$O(\text{poly } n)$	$O((\log^3 n)(\log \log n) \log^* n)$

Bibliography

- John Augustine, William K. Moses Jr., Gopal Pandurangan. **Distributed MST Computation in the Sleeping Model : Awake-Optimal Algorithms and Lower Bounds**. In PODC 2022 (BA).
- Alkida Balliu, Juho Hirvonen, Darya Melnyk, Dennis Olivetti, Joel Rybicki, Jukka Suomela. **Local Mending**. In SIROCCO 2022.
- Leonid Barenboim, Tzalik Maimon. **Deterministic Logarithmic Completeness in the Distributed Sleeping Model**. In DISC 2021.
- Fabien Dufoulon, William K. Moses Jr., Gopal Pandurangan, **Distributed MIS in O(log log n) Awake Complexity**. In PODC 2023.
- Guy Even, Moti Medina, Dana Ron. **Best of two local models : Centralized local and distributed local algorithms** In Inf. Comput. 262, 2018.
- Mohsen Ghaffari, Fabian Kuhn, Yannic Maus. **On the complexity of local distributed graph problems**. In STOC 2017.
- Will Rosenbaum, Jukka Suomela. **Seeing far vs. seeing wide** In PODC 2020.