



INSTITUT  
DE RECHERCHE  
EN INFORMATIQUE  
FONDAMENTALE

# Redundancy in Distributed Verification

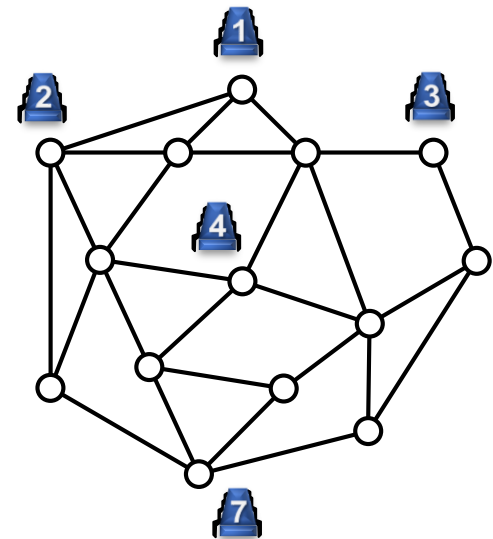
Ami Paz

Joint work with:

Laurent Feuilloley, Pierre Fraigniaud, Juho Hirvonen, Mor Perry

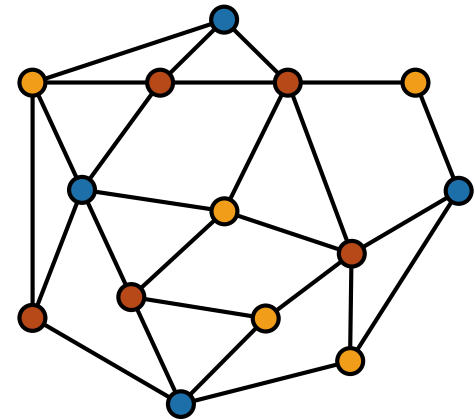
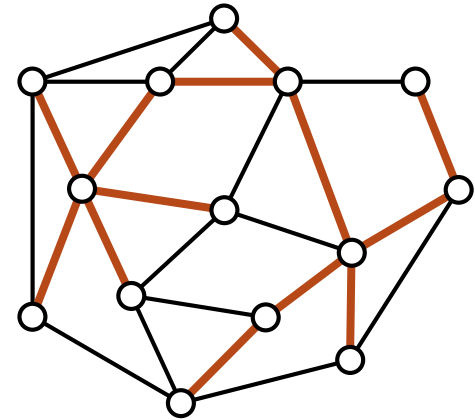
# Distributed Verification

- Graph  $G = (V, E)$ 
  - Node = processor
  - Edge = communication link
- Synchronous message passing, no faults
- Nodes have unique ids:  $1, \dots, n$
- No central control
- No shared memory



# Distributed Verification

- Configuration:  $(G, x)$
- Predicates  $P$  of a graph  $G$ :
  - $\text{Diam}(G) < 4$
  - $G$  is 3-colorable
- Predicates  $P$  of a configuration  $(G, x)$ :
  - $x$  is a spanning tree
  - $x$  is a proper coloring



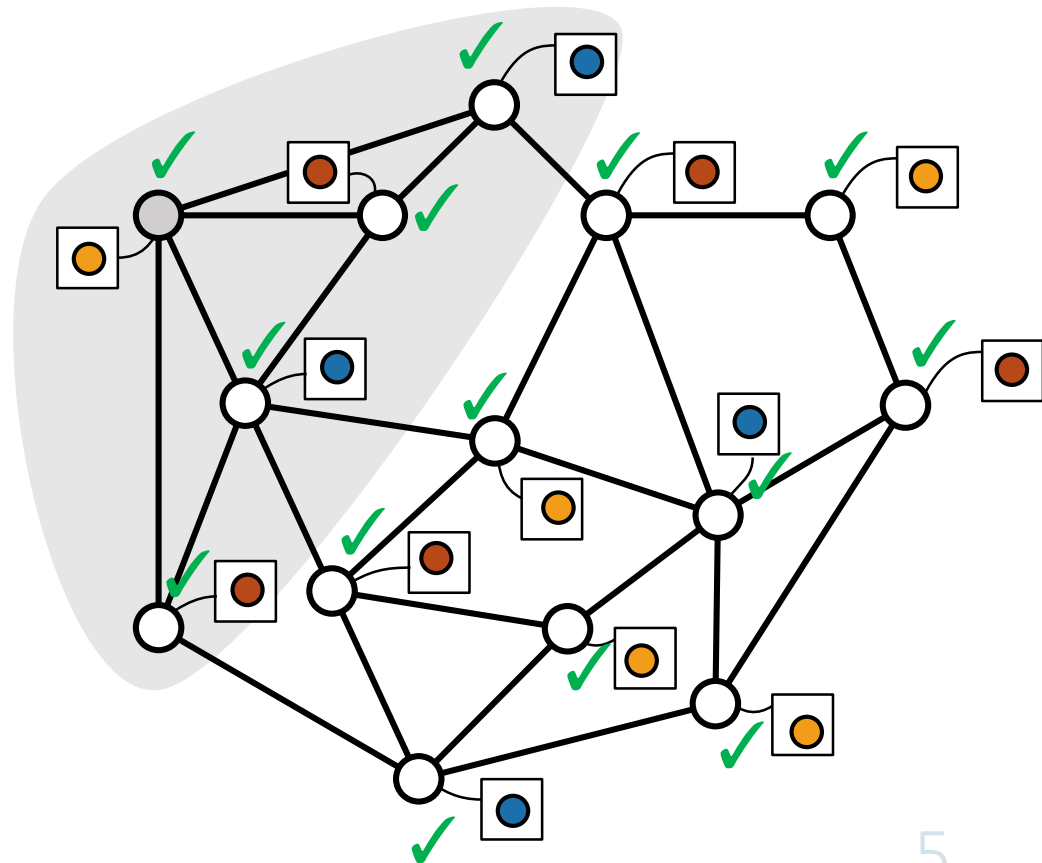
# Decision Mechanism

- Claim:  $(G, x)$  satisfies predicate  $P$
- Prover: assigns labels to nodes
- Verification process - each node:
  - Collects labels from neighbors
  - Decides by the labels
- $P$  holds  $\iff$  all nodes accept
- proof-size = maximum label size

Goal:  
Minimize the proof size

# Example: 3-colorability

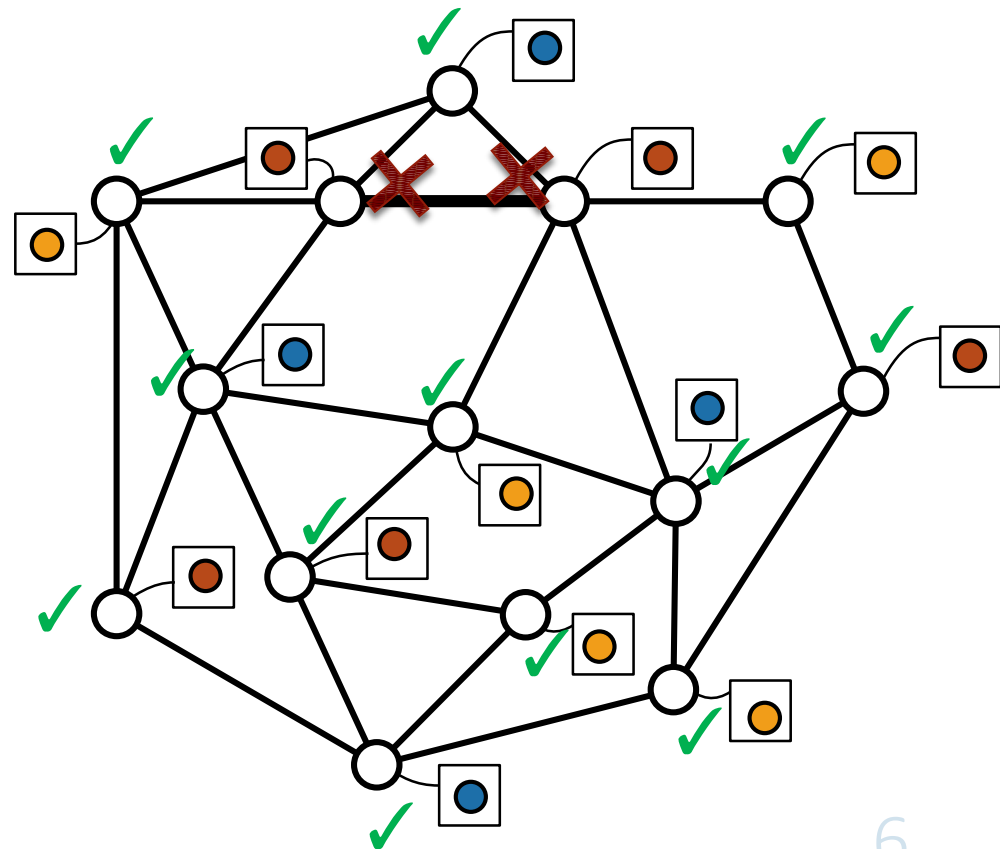
- Claim: The graph is 3-colorable
- Prover: assigns colors
- Verification:  
all neighbors' colors  
different from mine



# Example: 3-colorability

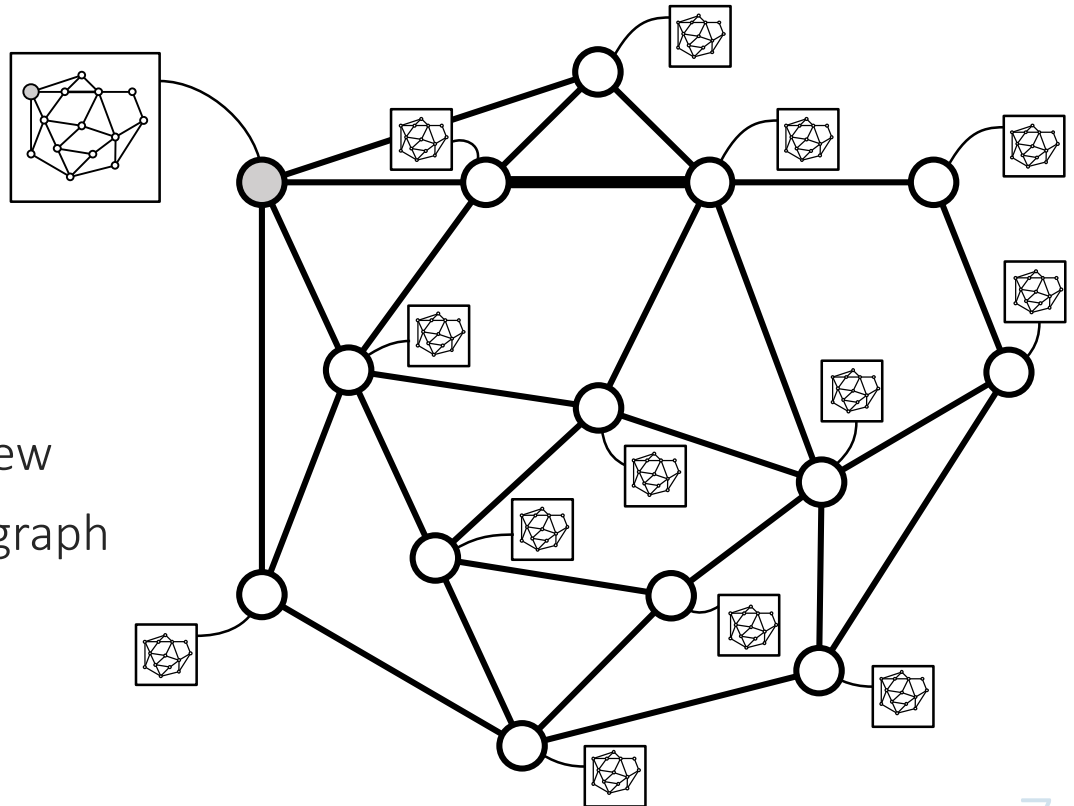
- Claim: The graph is 3-colorable

What if it is not 3-colorable?



# Example: 3-colorability

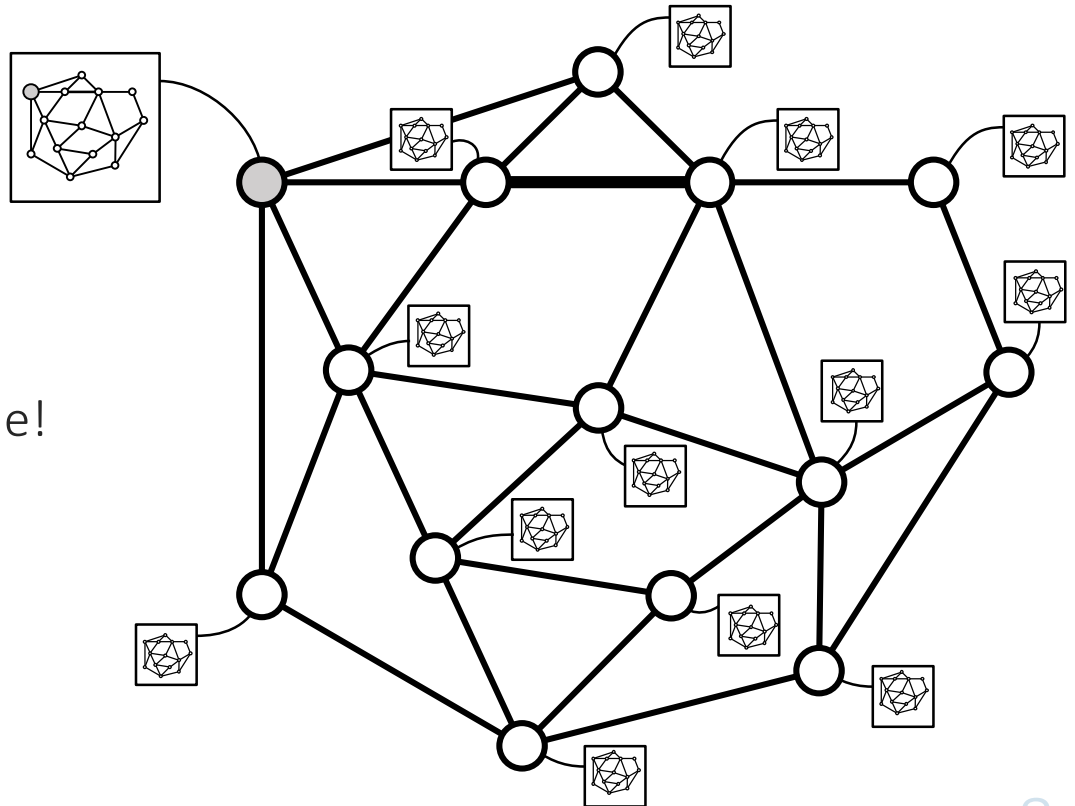
- Claim: The graph is **not** 3-colorable
- Prover:  
Give each node the whole graph
- Verification:
  - all labels equal
  - Consistent with node's view
  - Contain non 3-colorable graph



# Example: 3-colorability

- Claim: The graph is **not** 3-colorable

- The **universal** scheme
  - Takes  $\Theta(n^2)$  bits
  - No better scheme possible!





# Why Should we Care?

- Practical: Self stabilization
  - Make sure a network is in a good state
- Theoretical: Non-determinism in distributed graph algorithms
  - A distributed analogue of NP

# Previous Work

0

- LCL [Naor, Stockmeyer 93]

$\Theta(1)$

- Colorability

$\Theta(\log n)$

- Spanning tree [Korman, Kutten, Peleg 05]

$\Theta(\log^2 n)$

- Minimum spanning tree [Korman, Kutten 06]

$\tilde{\Theta}(n)$

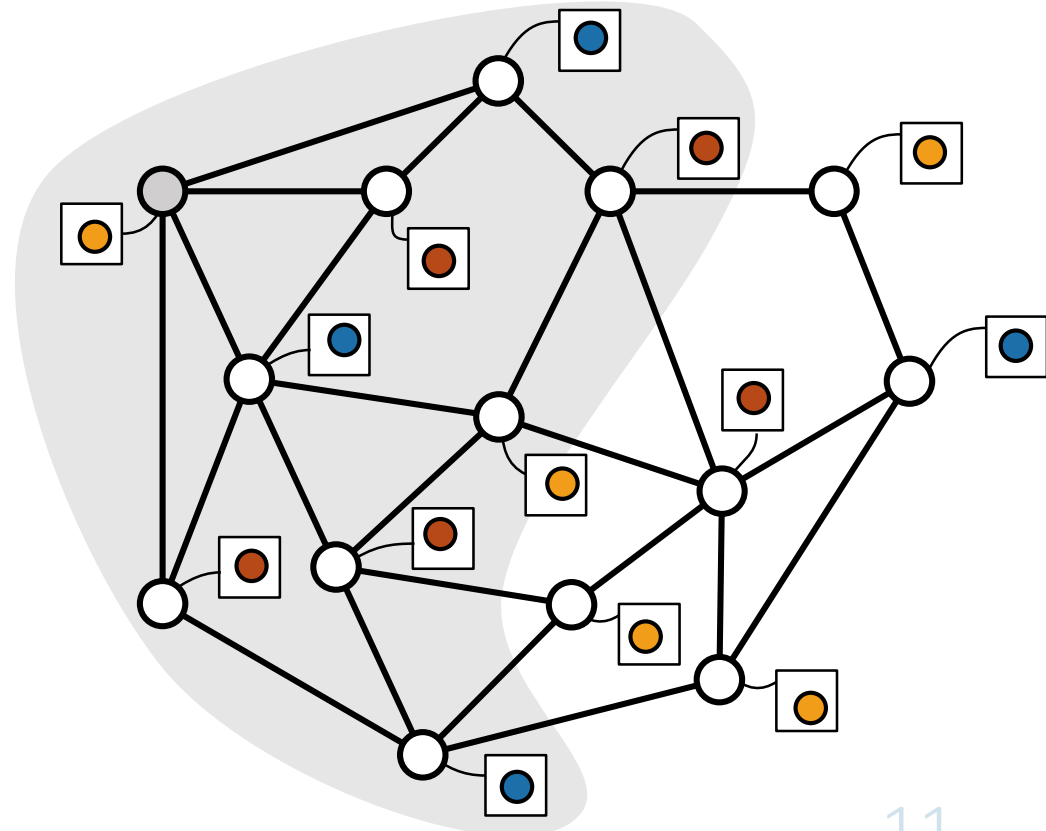
- Diameter [Censor-Hillel, Paz, Perry 17]

$\tilde{\Theta}(n^2)$

- Non-3-colorability [Göös, Suomela 11]

# Distance- $t$ Verification

- Nodes decide by their  $t$ -neighborhood



# Distance- $t$ Verification

- Nodes decide by their  $t$ -neighborhood
- Scaling [Ostrovsky, Perry, Rosenbaum 17]
  - $f(t)$ -scaling:

$$\text{proof-size}(n, t) = O\left(\frac{\text{proof-size}(n, 1)}{f(t)}\right)$$

# Previous Results

$t$ -scaling (linear)

- The universal scheme [Ostrovsky, Perry, Rosenbaum 17]
- Minimum spanning tree, for  $t = \log n$   
[Korman, Kutten, Masuzawa 15]
- That is:  $\log n$ -bits for distance- $\log n$   
(instead of  $\log^2 n$ -bits for distance-1)

# Upper Bounds

$t$ -scaling (linear)

- Any predicate on cycles, paths, grids and trees
- Diameter, spanning tree, minimum spanning tree, spanners

$b(t)$ -scaling (optimal)

- The universal scheme

$b(t)$  = minimum size  
of a ball of radius  $t$

# Lower Bounds

- Diameter and additive spanners:
  - Any distance- $t$  scheme has size  $\tilde{\Omega}\left(\frac{n}{t}\right)$
  - That is, the linear-scaling scheme is optimal
- New lower-bound technique

# Upper Bounds

Schemes for Distance- $t$  Verification



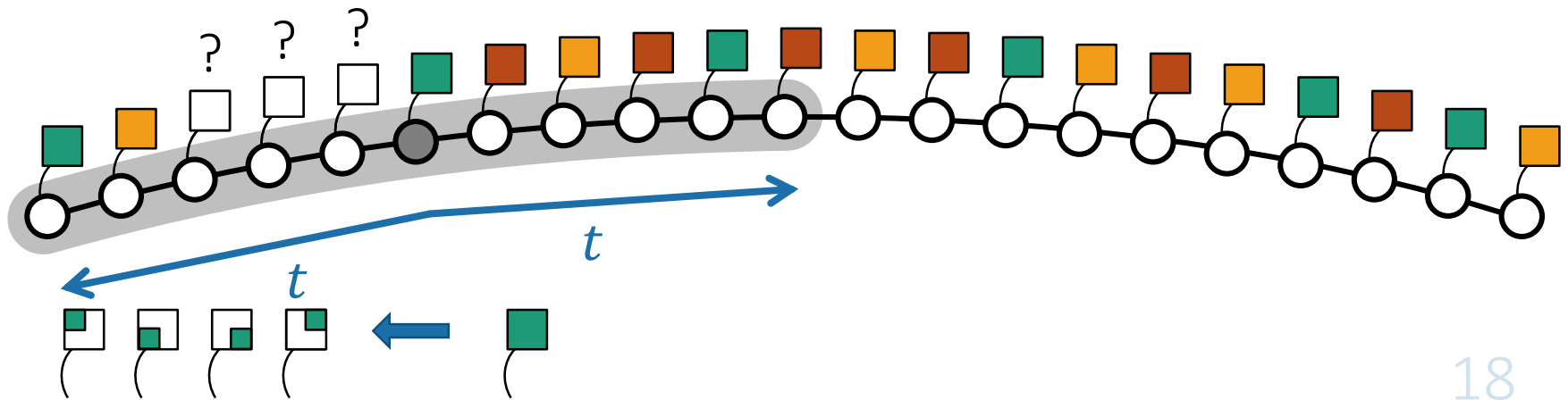
# Linear Scaling on Specific Graphs

- Any verification scheme scales linearly on:
  - Cycles
  - Grids
  - Trees
  - And more...

# Scaling on a Cycle

Start with a distance-1 scheme for a cycle

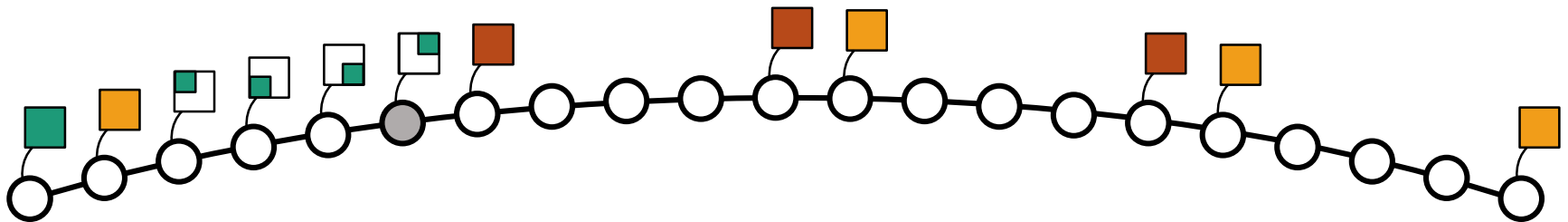
- Keep 2 out of every  $t$  labels
  - Reduces *average* label size
- Verification:
  - Simulate all possible assignments to the missing labels



# Scaling on a Cycle

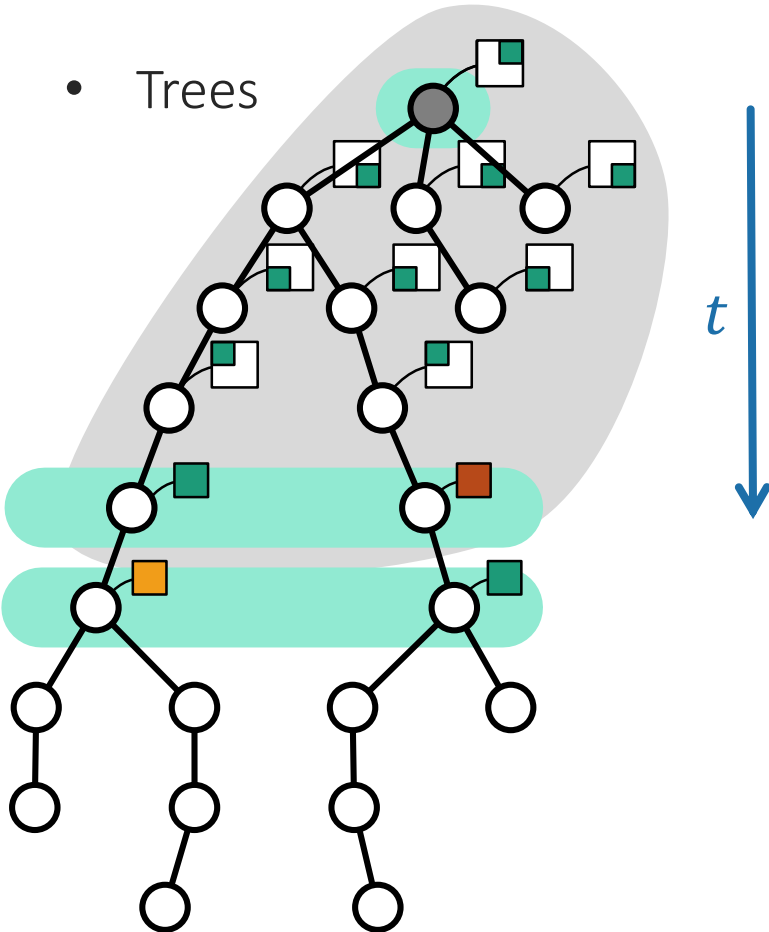
Given a sparse scheme

- Spread the labels among the neighbors without labels
  - Reduces maximum label size
- Verification:
  - Reconstruct the labels, and proceed as before

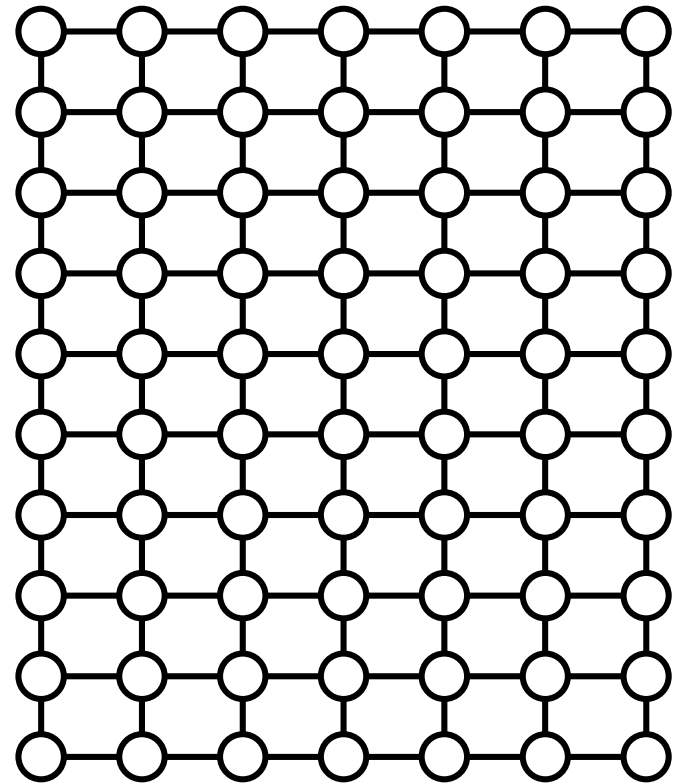


# Other Graphs

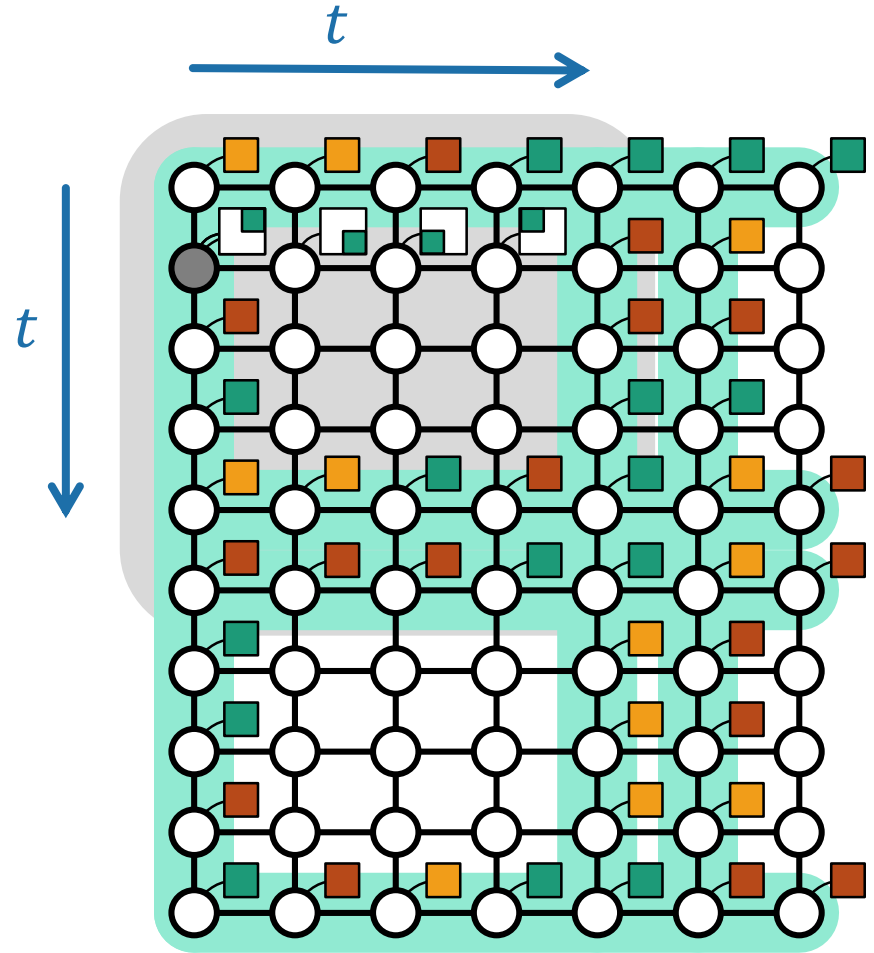
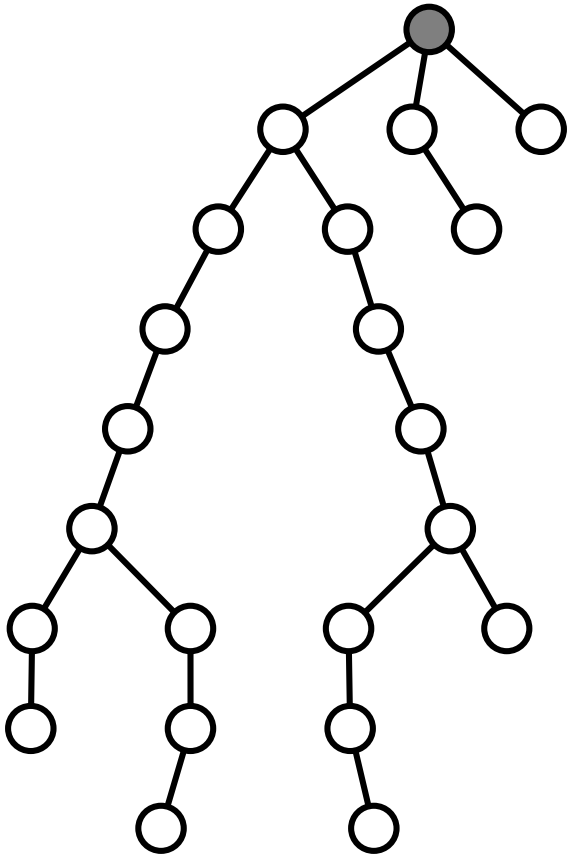
- Trees



- Grids

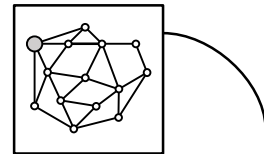


# Other Graphs



# The Uniform Scheme Scale Optimally

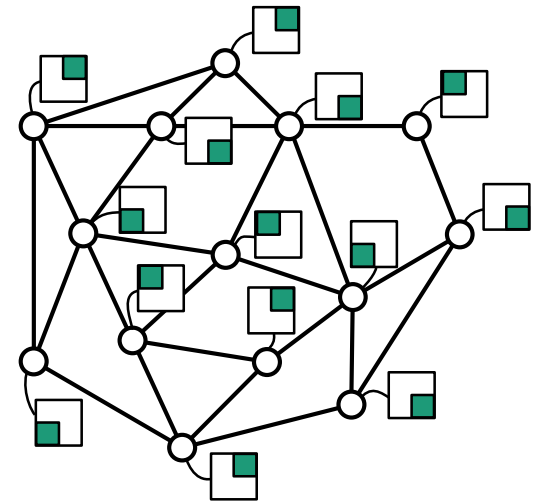
- Uniform Scheme: all labels are the same
- The universal scheme:  $O(n^2)$ -bit labels
  - Works for any predicate
- Optimal for:
  - Non 3-colorability
  - Symmetric graphs
  - ...



# Scaling of Uniform Labels

- Split the label into  $\sim b(t)$  blocks
- Assign random blocks to each node
- Claim:

For any node,  
all blocks exist in its  
 $t$ -neighborhood whp



- Reconstruct the labels and use distance-1 verification

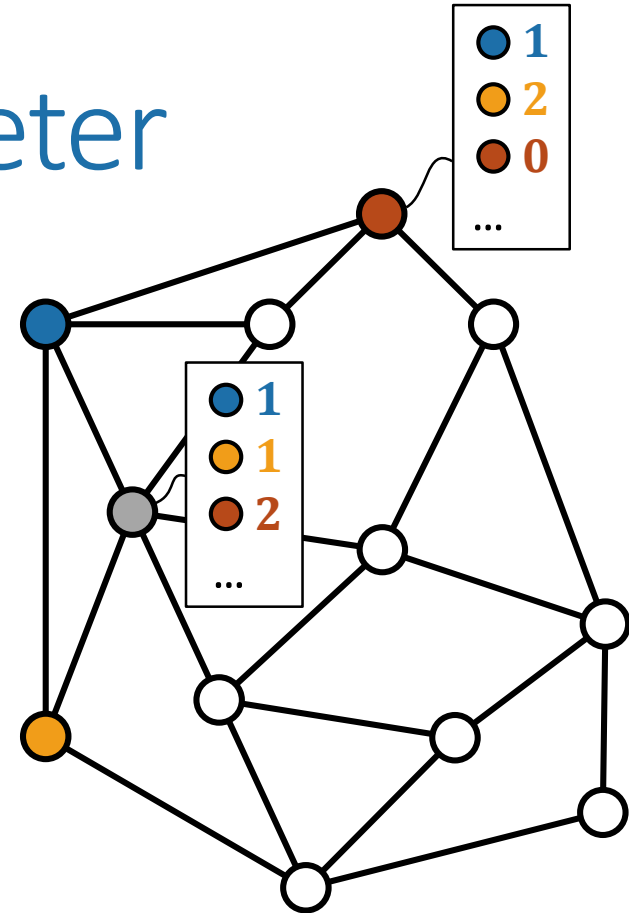
# Uniform Labels – Afterthoughts

- Need to enumerate blocks – lose log factor
- We use the probabilistic method for the proof; the scheme and verification are **deterministic**



# Linear Scaling for Diameter

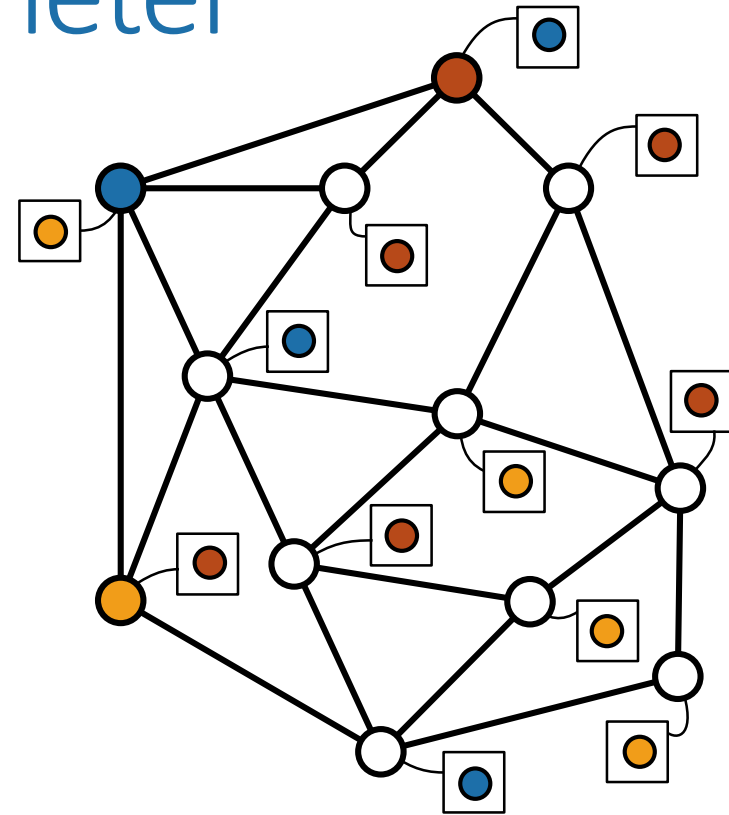
- Diameter
  - Distance-1: Must keep all distances



# Linear Scaling for Diameter

- Diameter
  - Distance-1: Must keep all distances
  - Distance- $t$ : Keep distances from  $\sim \frac{n}{t}$  nodes
- Claim:

For any source,  
on any sub path of length  $t$   
of the BFS tree from it,  
at least one label is kept



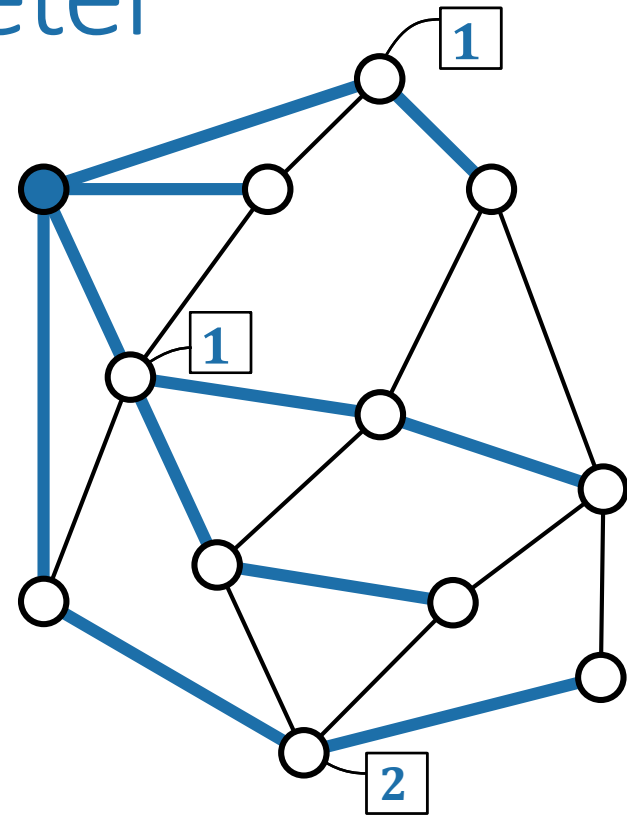
# Linear Scaling for Diameter

- Diameter
  - Distance-1: Must keep all distances
  - Distance- $t$ : Keep distances from  $\sim \frac{n}{t}$  nodes

• Claim:

For any source,  
on any sub path of length  $t$   
of the BFS tree from it,  
at least one label is kept

- Spanners
  - Similar, but keep distance both for the graph and the spanner



# Linear Scaling for Other Predicates

- Spanning tree
  - Labels of size  $O\left(\frac{\log n}{t}\right)$
- MST
  - Labels of size  $O\left(\frac{\log^2 n}{t}\right)$  for  $t = O(\log n)$
  - Optimal for  $t = \Theta(\log n)$

# Lower Bounds

When Alice and Bob are Far Apart

# The Disjointness( $k$ ) Problem

- Two players: Alice and Bob
- Inputs
  - Alice  $S_A \subseteq [k]$
  - Bob  $S_B \subseteq [k]$
- Goal: Does  $S_A \cap S_B = \emptyset$ ?

*Alice*

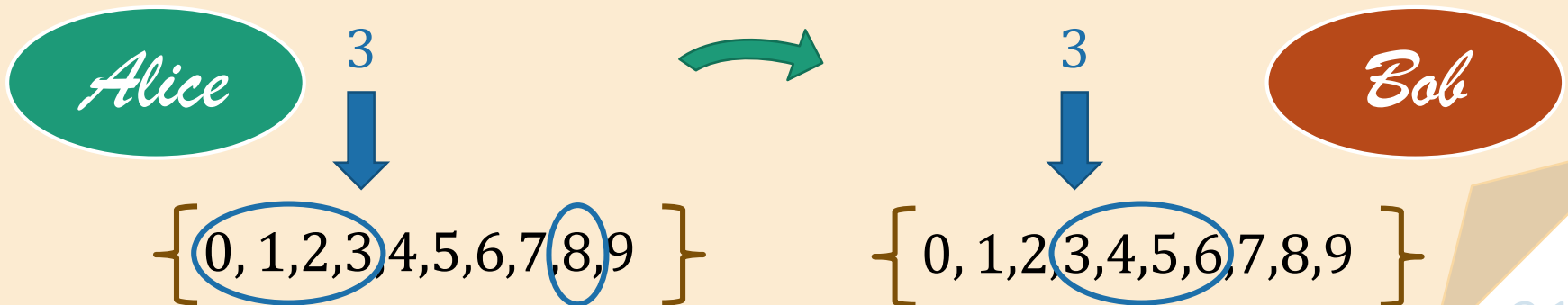
{ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 }

*Bob*

{ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 }

# Non-deterministic Disjointness

- Alice and Bob get a *hint*
- Example: For non-disjointness, hint = index of intersection
  - Communicating  $\log k$  bits is enough
- For *disjointness*, need to communicate  $\Omega(k)$  bits



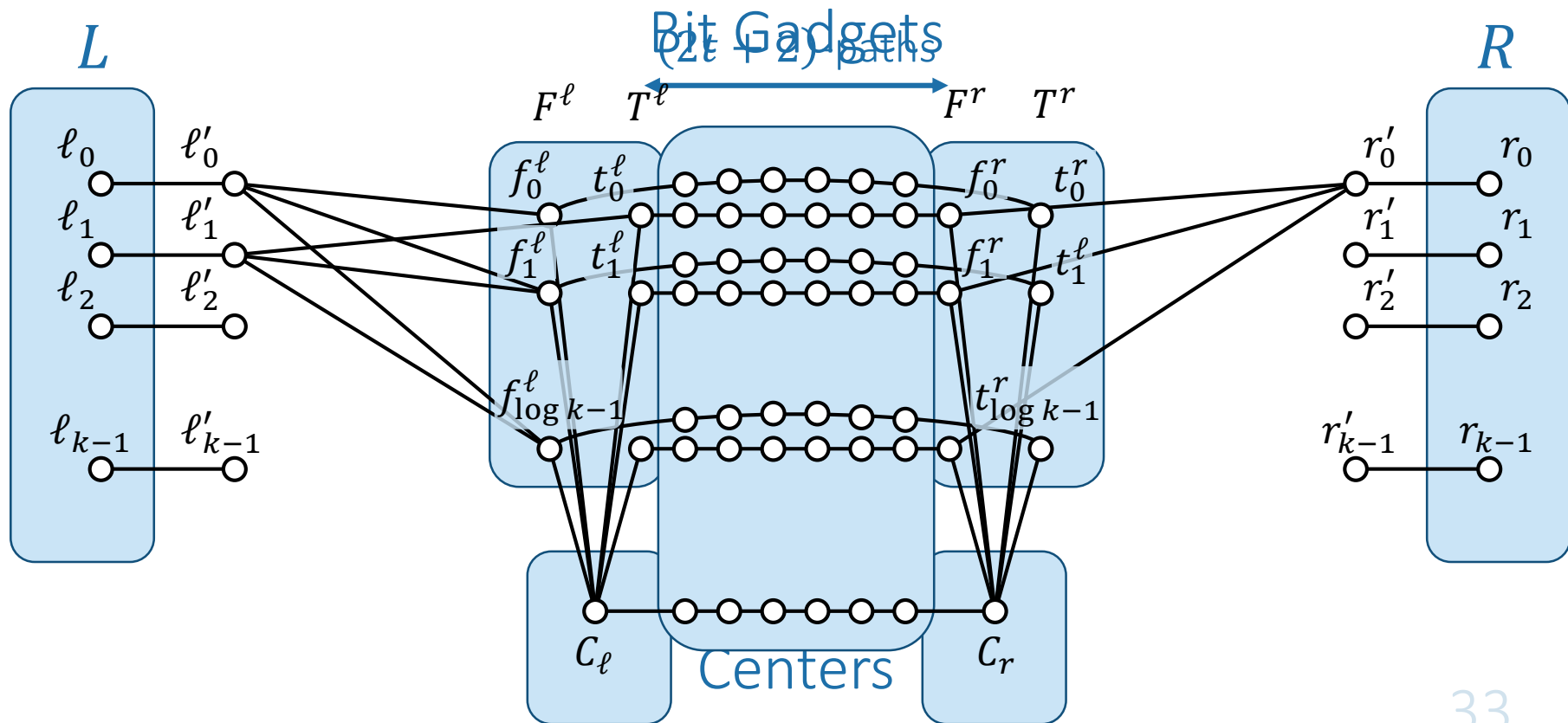
# Verifying Diameter

- Linear scaling is optimal
- Assume the contrary:
  - there is a scheme labels of  $\tilde{O}\left(\frac{n}{t}\right)$  bit
- Alice and Bob are **experts in distributed verification!**
  - Use this scheme to solve disjointness



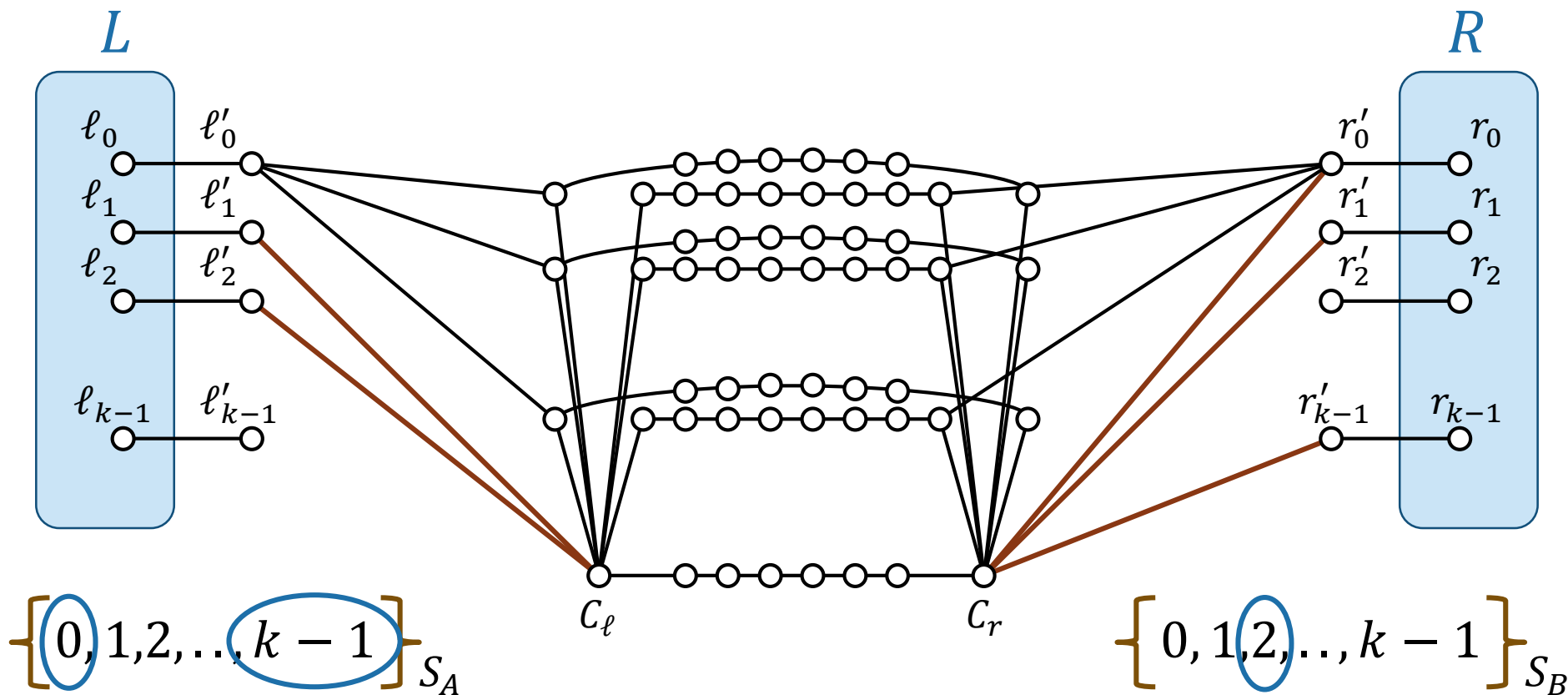
# The Graph

$$k \approx \frac{n}{4}$$



# The Graph – Adding Edges

- Edges depending on  $S_A, S_B$

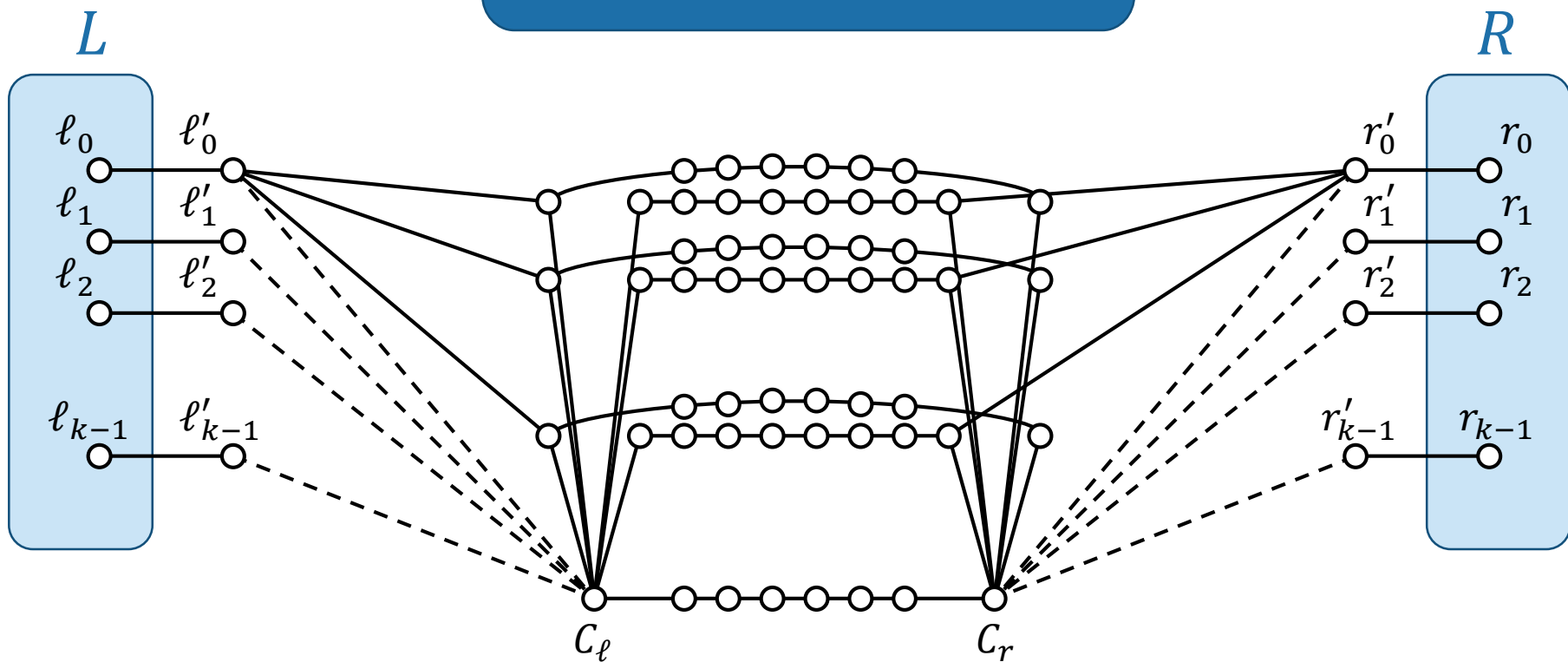


# The Graph – Diameter

- Diameter implies

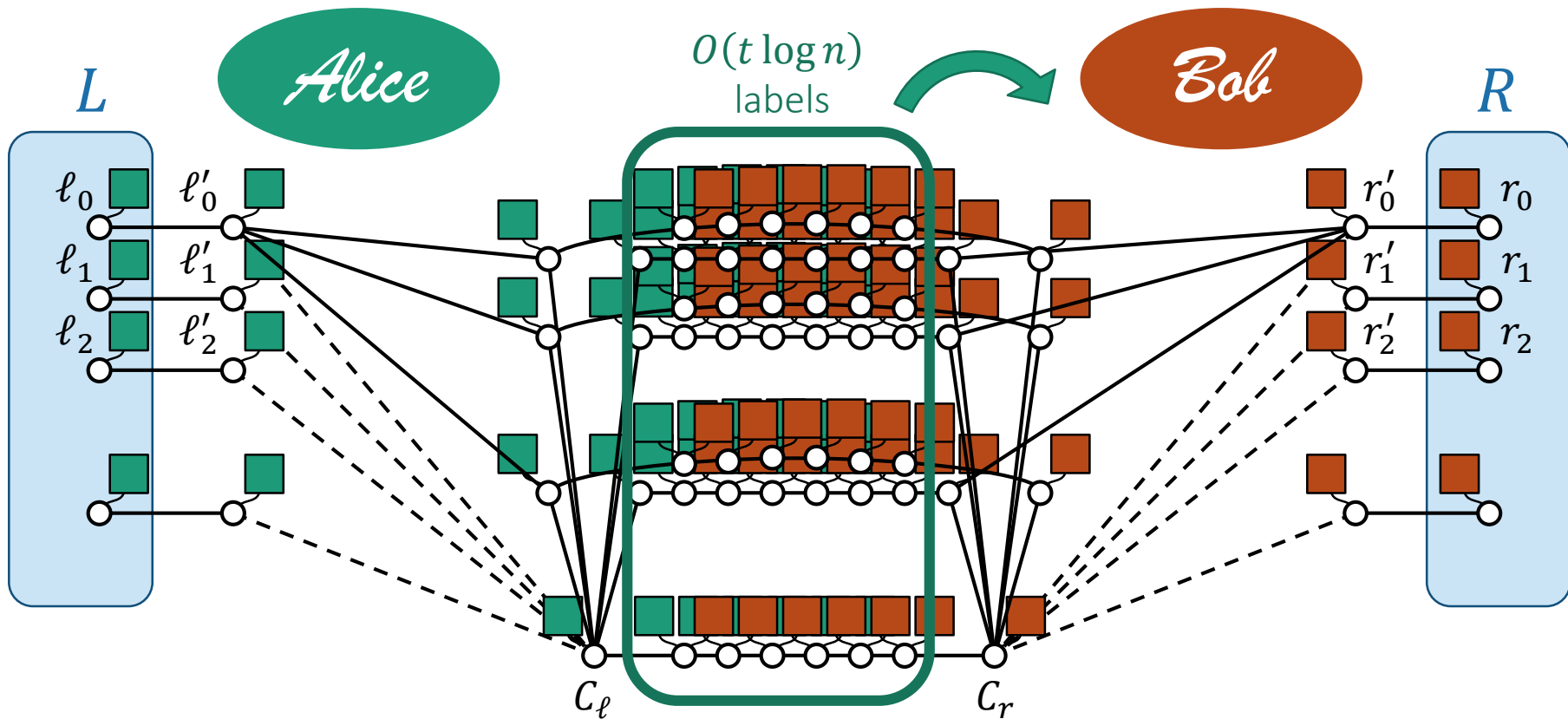
$$S_A \cap S_B = \emptyset \Rightarrow \text{Diam} = 2t + 7$$

$$S_A \cap S_B \neq \emptyset \Rightarrow \text{Diam} = 2t + 8$$



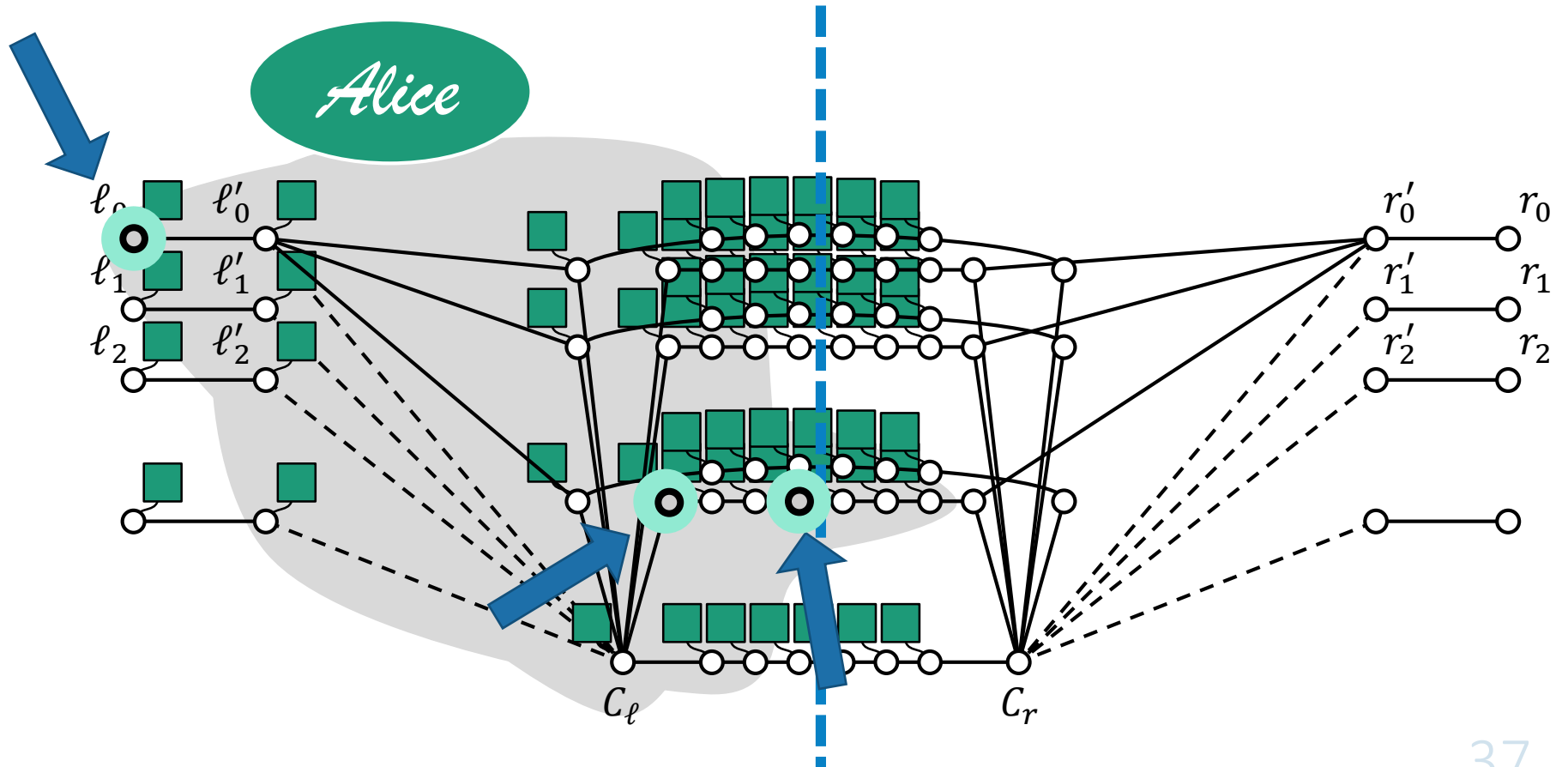
# The Simulation

- Interpret non-deterministic strings as labels



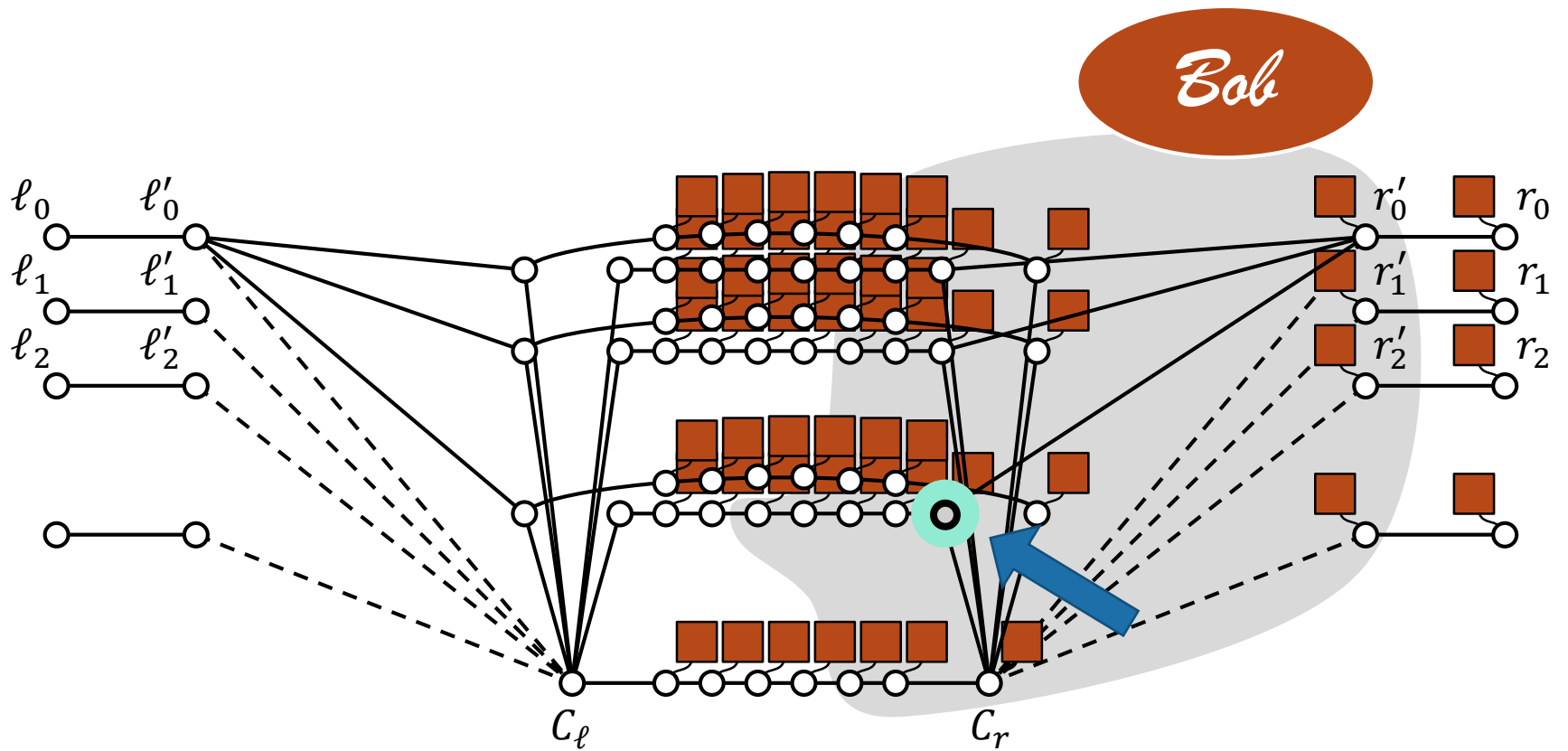
# The Simulation

- Simulate verification



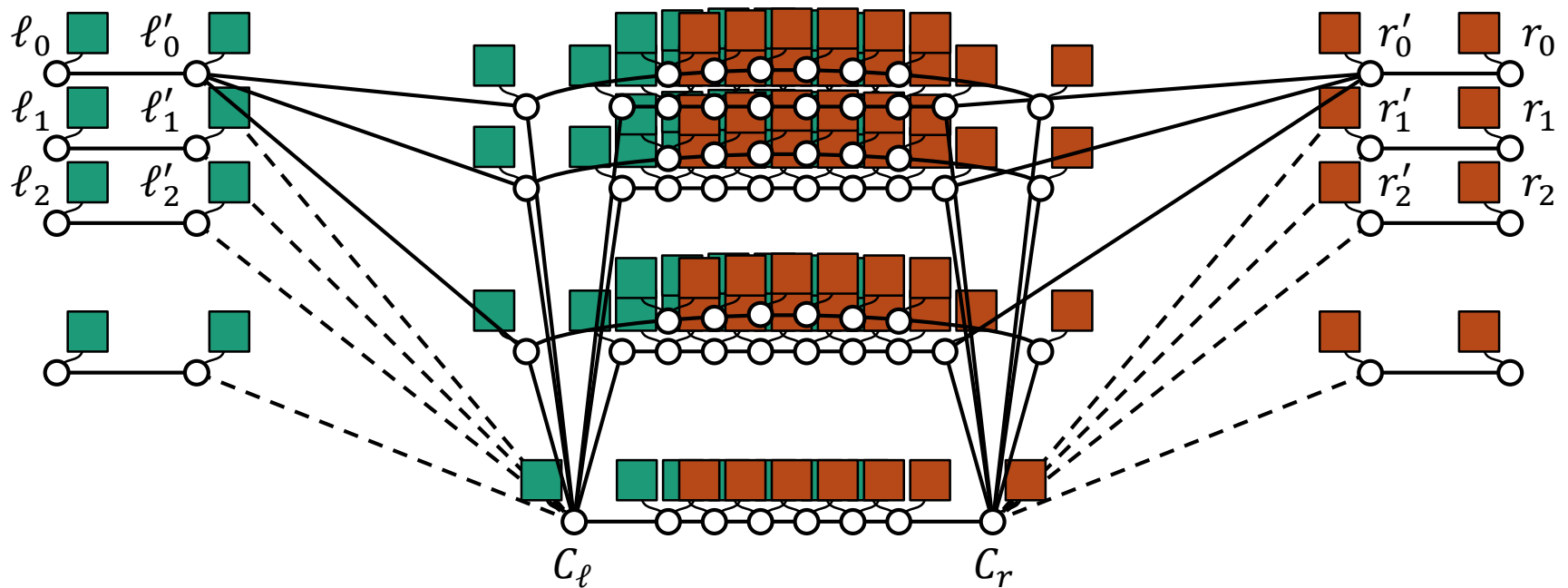
# The Simulation

- Simulate verification



# The Simulation

- Decide  $\text{Diam} = 2t + 7$  ?



# Deciding Disjointness

- Decide  $\text{Diam} = 2t + 7$  ?

$$S_A \cap S_B = \emptyset$$

$$\Rightarrow \text{Diam} = 2t+7$$

$\Rightarrow$  All nodes accept

$\Rightarrow$  Alice and Bob return “disjoint”



# Deciding Disjointness

- Decide  $\text{Diam} = 2t + 7$  ?

$$S_A \cap S_B \neq \emptyset$$

$$\Rightarrow \text{Diam} = 2t+8$$

$\Rightarrow$  Some node **rejects**

$\Rightarrow$  Alice or Bob return “Not-disjoint”

# To Conclude

- Alice and Bob can decide disjointness
- Exchange  $O(t \log n)$  labels
- Must exchange  $\Omega(n)$  bits
- There is a label of  $\Omega\left(\frac{n}{t \log n}\right)$ 
  - Scaling lower bound

# Lower Bound – Afterthoughts

- A scheme for diameter cannot scale better than  $\Omega\left(\frac{n}{t \log n}\right)$
- Technique:
  - Non-deterministic communication complexity
  - Small-cut construction
  - Simulation technique from CONGEST model
  - But, our model resembles the LOCAL model

# Open Questions

- Is linear scaling always possible?
  - Proved for specific graphs: cycles, trees,...
  - Proved for specific predicates: spanning tree, MST, diameter
- Is optimal scaling always possible?
  - Proved for all predicates that require the universal scheme

*Thank You*