

# **Redundancy in distributed proofs**

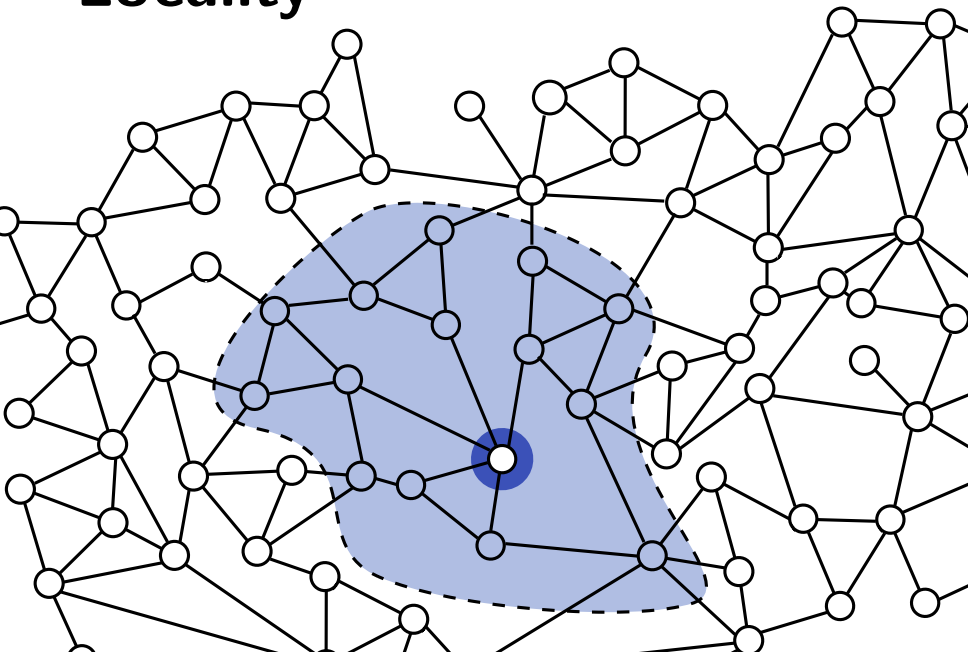
Laurent Feuilloley

Université Paris Diderot

based on joint works with Pierre Fraigniaud,  
Juho Hirvonen, Ami Paz, and Mor Perry

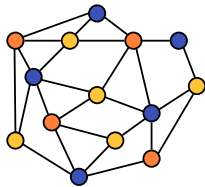
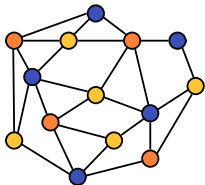
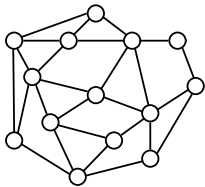
ANR Descartes - 28 mars 2018 - IRIF

# Locality



# **Distributed decision**

# Building vs. deciding



Yes/No

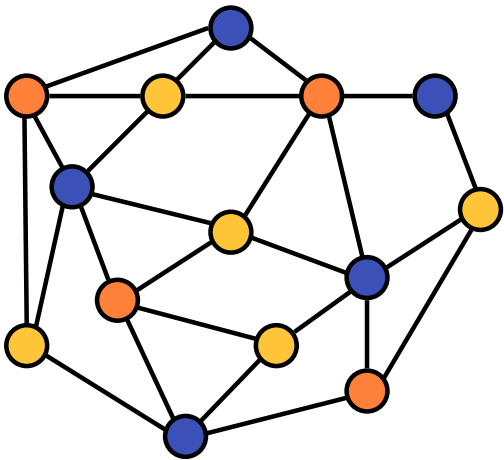
# Distributed languages

Context :

- ▶ Communication graph  $G$
- ▶ Node inputs,  $x : v \mapsto x(v)$

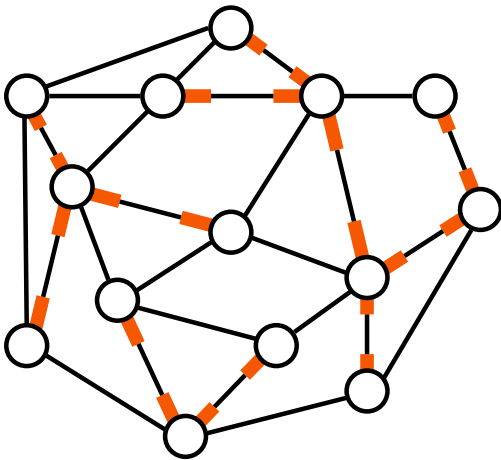
A language is a set  $\{(G, x)\}$ .

# Properly-colored graphs



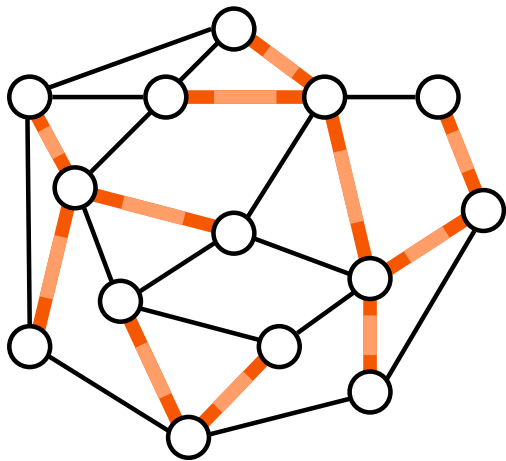
$$\mathcal{L} = \{(G, x) \text{ s.t. } x \text{ is a proper coloring of } G\}$$

# Spanning forest



$$\mathcal{L} = \{(G, x) \text{ s.t. } x \text{ describes a spanning forest of } G\}$$

# Spanning forest



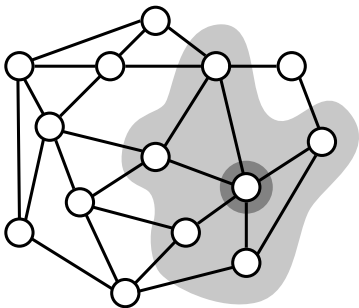
$$\mathcal{L} = \{(G, x) \text{ s.t. } x \text{ describes a spanning forest of } G\}$$



# Decision mechanism

Every node :

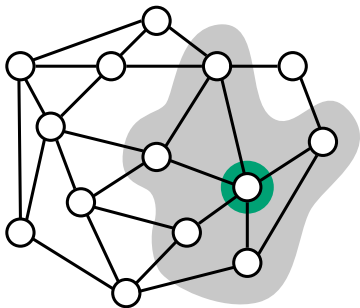
- ▶ gathers its 1-neighbourhood
- ▶ outputs a local decision **accept** or **reject**.



# Decision mechanism

Every node :

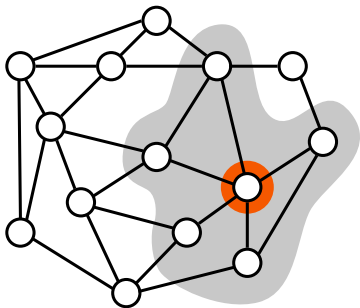
- ▶ gathers its 1-neighbourhood
- ▶ outputs a local decision **accept** or **reject**.



# Decision mechanism

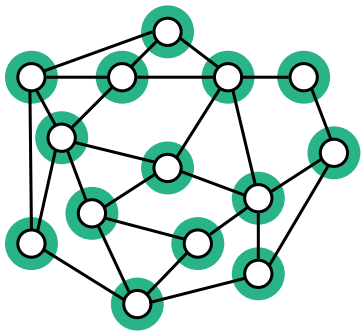
Every node :

- ▶ gathers its 1-neighbourhood
- ▶ outputs a local decision **accept** or **reject**.



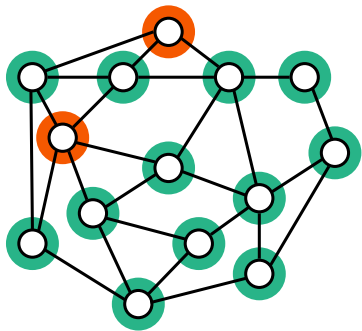
# Decision mechanism

$(G, x)$  is accepted  
if all node accept.

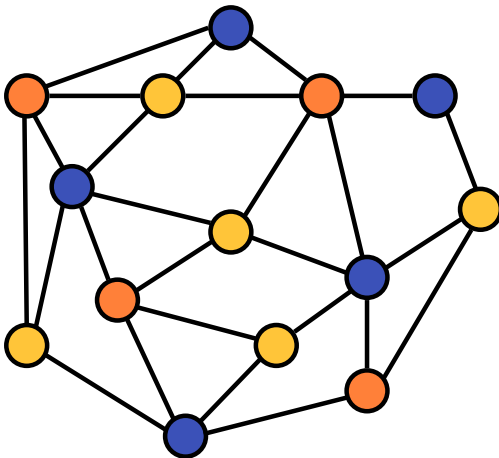


# Decision mechanism

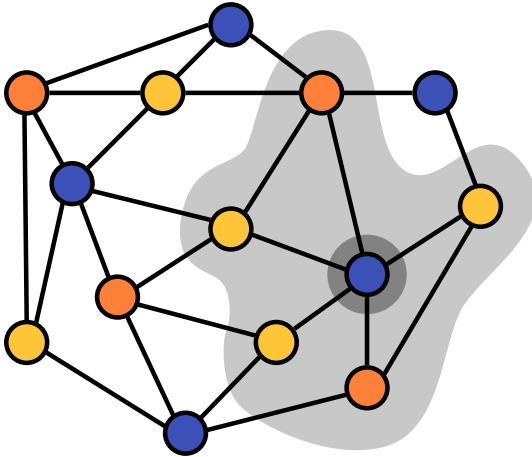
$(G, x)$  is rejected  
if at least one node  
rejects.



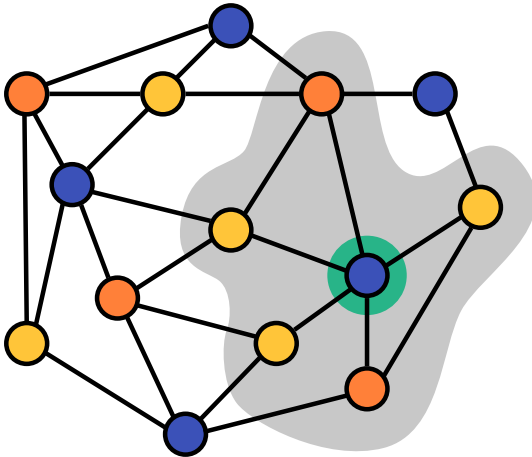
# Properly-colored graphs



# Properly-colored graphs

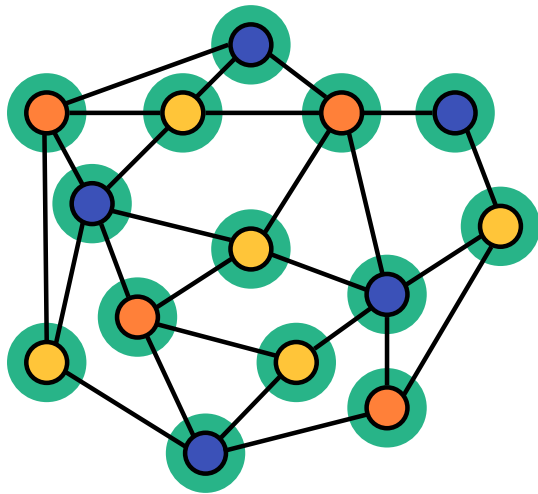


# Properly-colored graphs

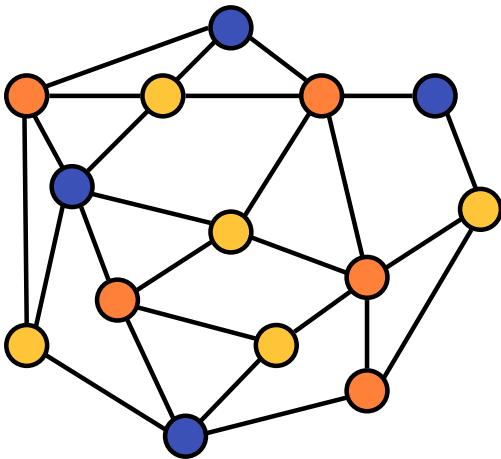




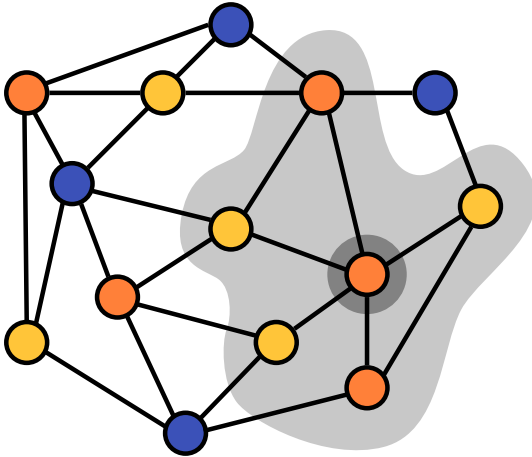
# Properly-colored graphs



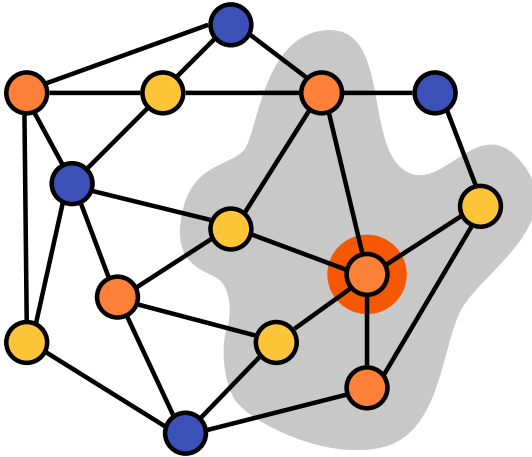
# Properly-colored graphs



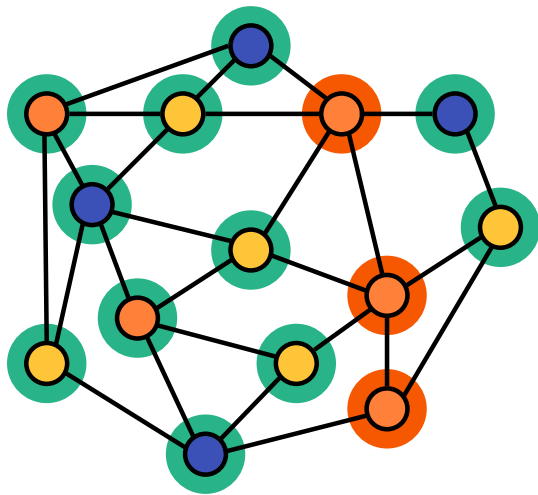
# Properly-colored graphs



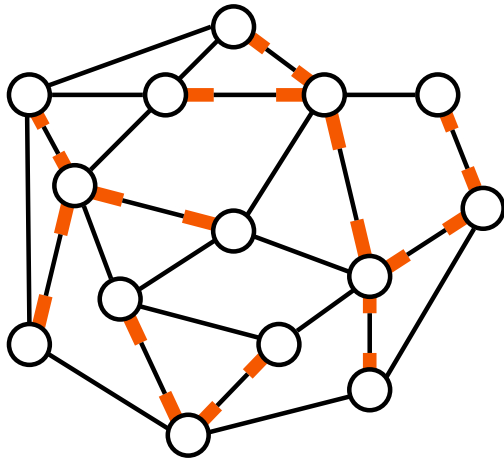
# Properly-colored graphs



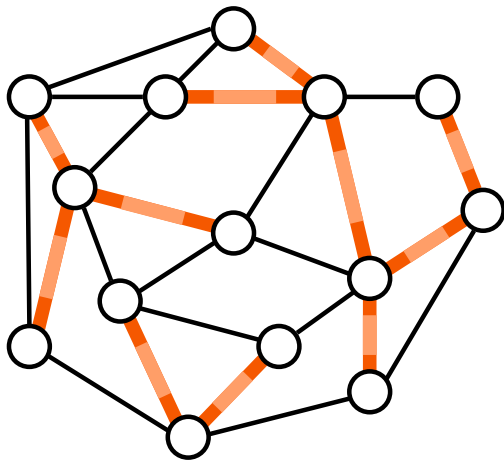
# Properly-colored graphs



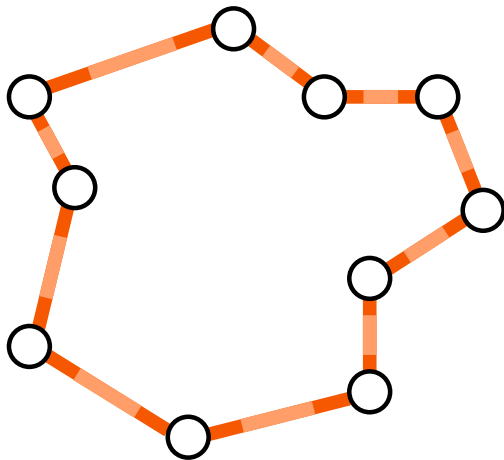
# Spanning forest



# Spanning forest

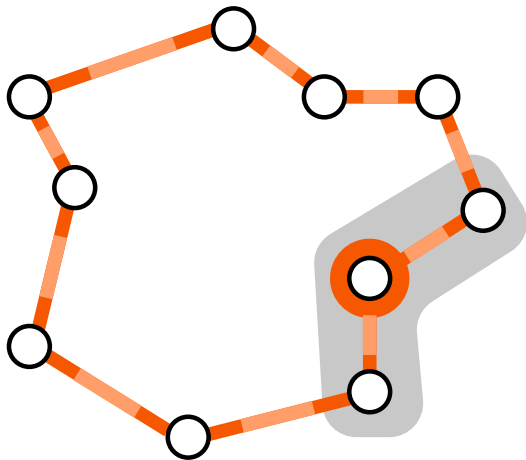


# Spanning forest

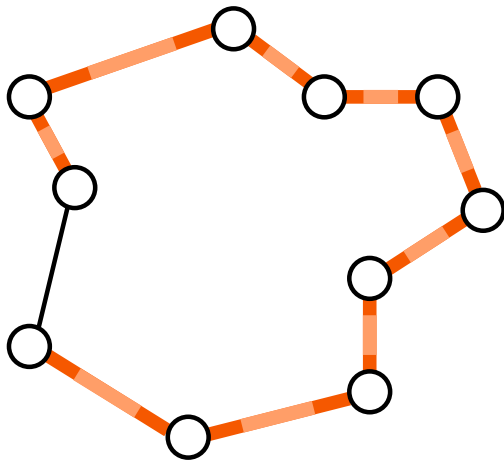




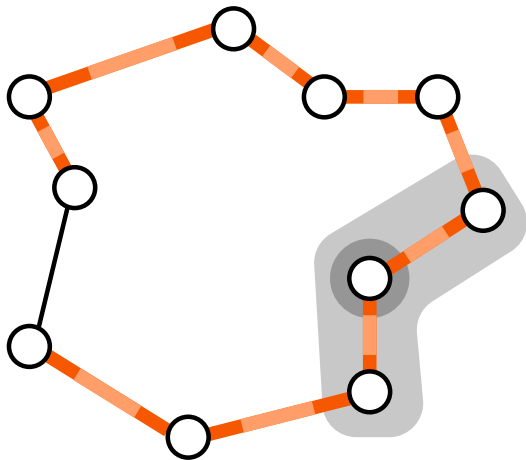
# Spanning forest



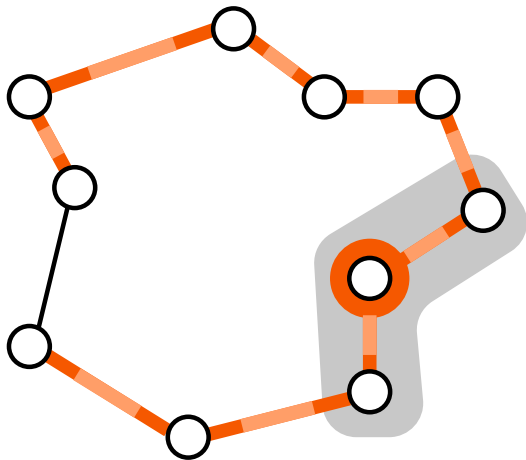
# Spanning forest



# Spanning forest



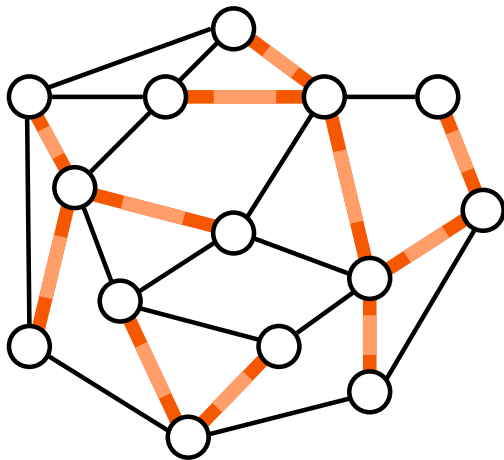
# Spanning forest



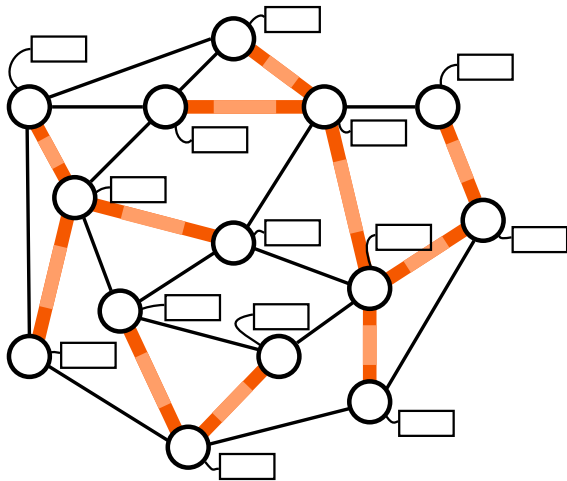
# **Proof-labeling schemes**

**Distributed  
non-determinism**

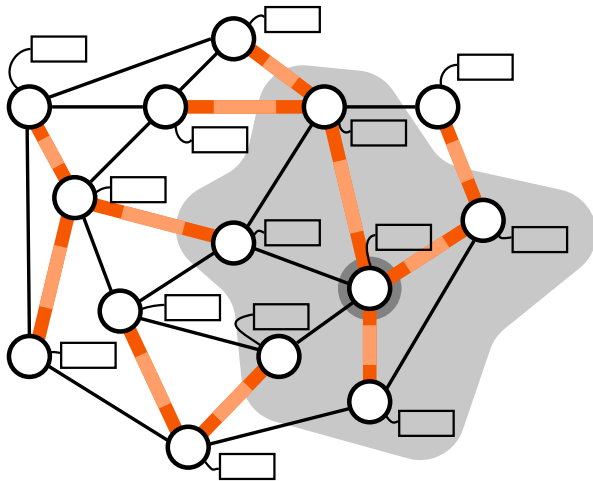
# Proof-labeling schemes



# Proof-labeling schemes



# Proof-labeling schemes





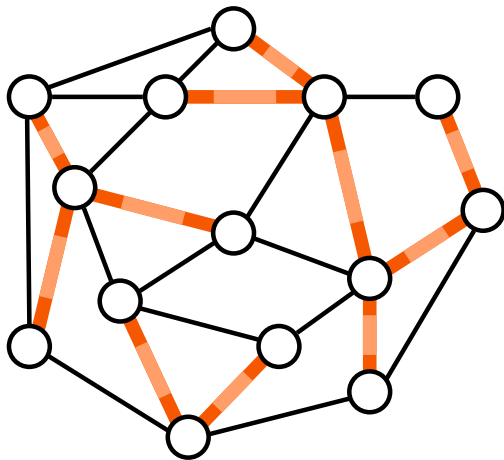
# Proof-labeling schemes

Given a proof-labeling scheme for  $\mathcal{L}$  :

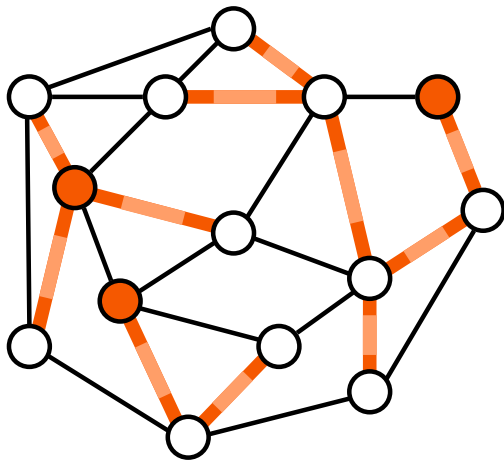
For all  $(G, x)$  :

- ▶ If  $(G, x) \in \mathcal{L}$  :  
 $\exists c$  s.t.  $(G, x, c)$  is **accepted**.
- ▶ If  $(G, x) \notin \mathcal{L}$  :  
 $\forall c$ ,  $(G, x, c)$  is **rejected**.

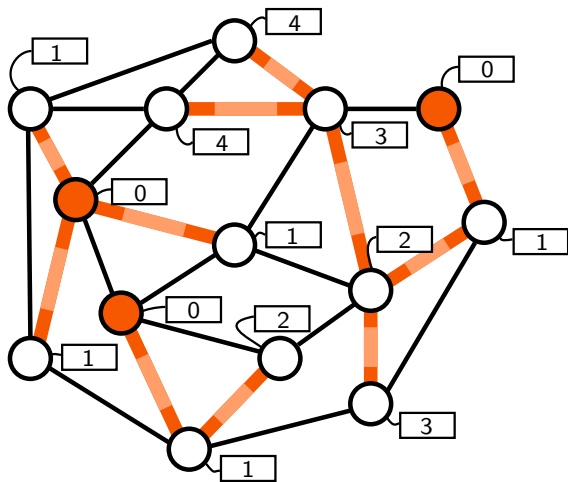
# PLS on spanning forest



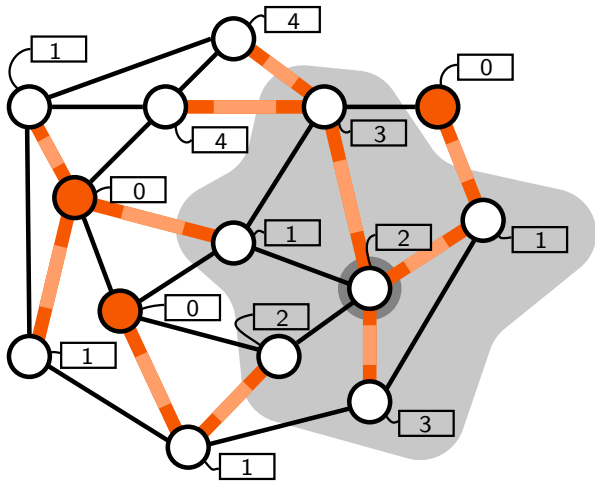
# PLS on spanning forest



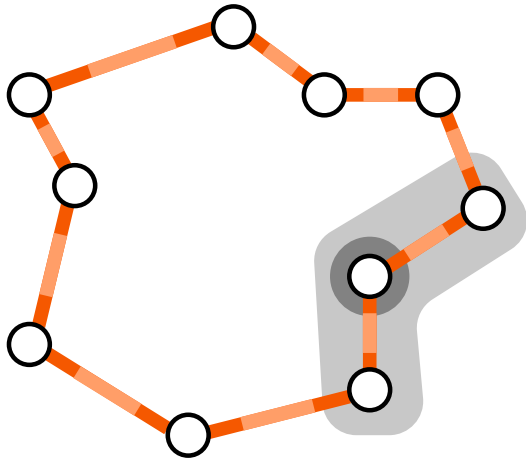
# PLS on spanning forest



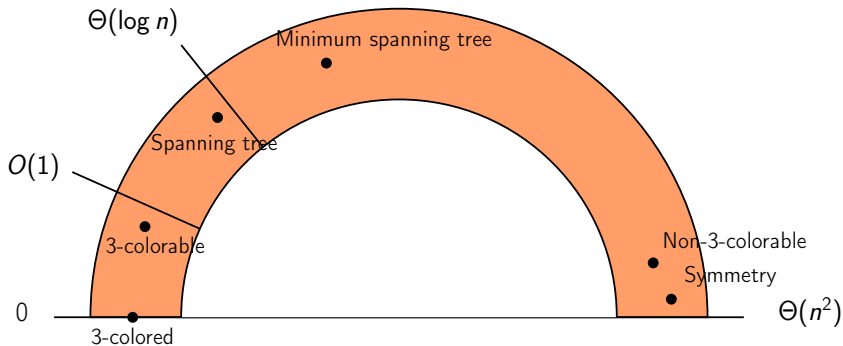
# PLS on spanning forest



# PLS on spanning forest



# Proof size



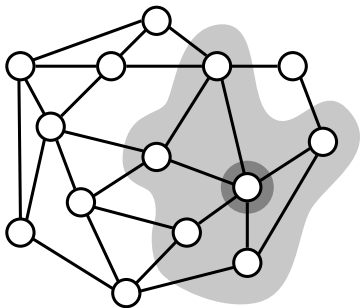
**Larger radius**



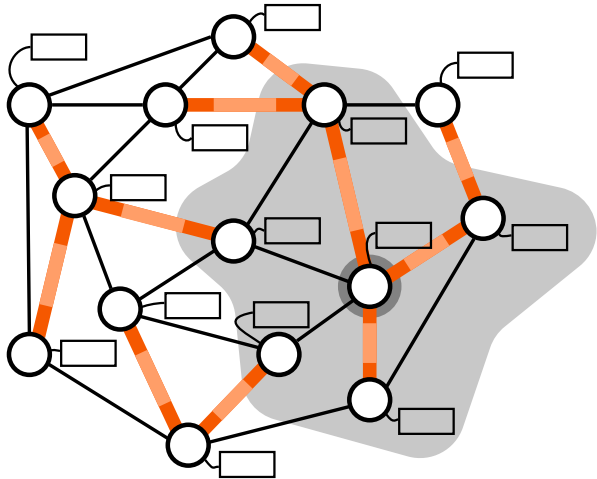
# Decision mechanism

Every node :

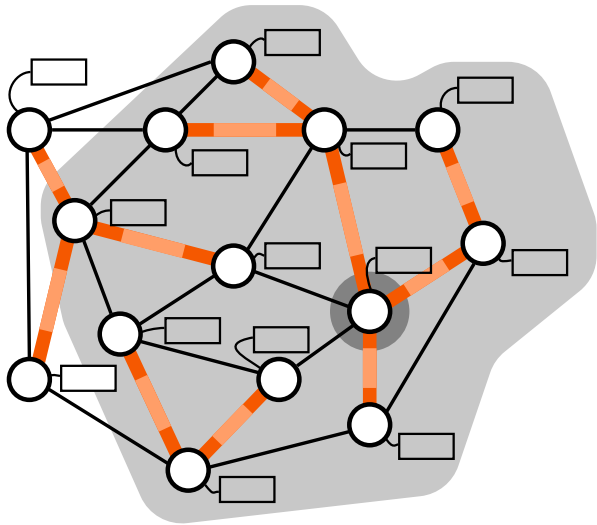
- ▶ gathers its  
1-neighbourhood
- ▶ outputs a  
local decision  
accept or reject.



# Larger radius



# Larger radius

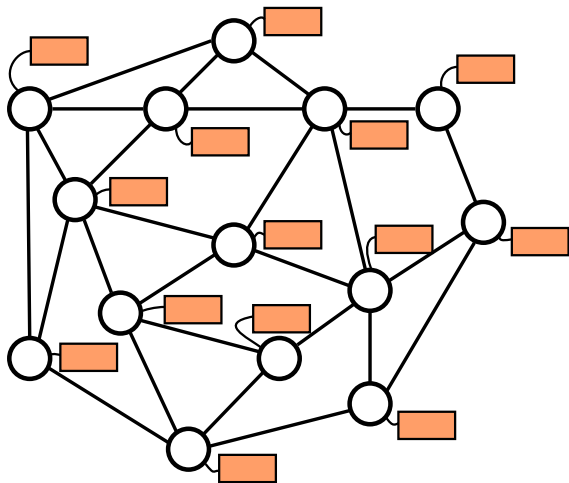


# Smaller proofs ?

What trade-offs between  
the radius  $t$  and the certificate size ?

- ▶ Can we always get  $s_t(n) = s_1(n)/t$  ?
- ▶ When can we get  $s_t(n) = s_1(n)/b(t)$  ?

# Spreading uniform proofs

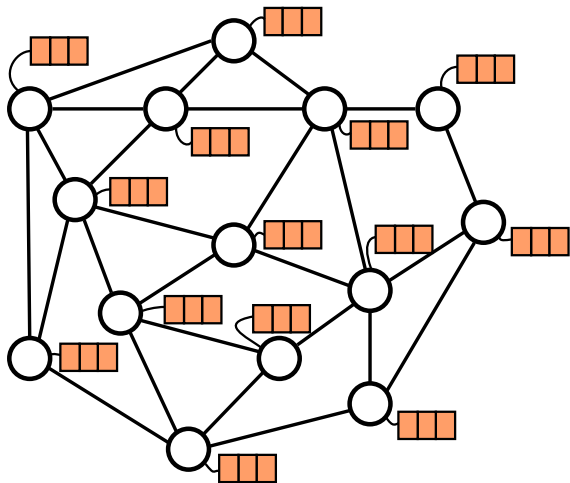


# Spreading uniform proofs

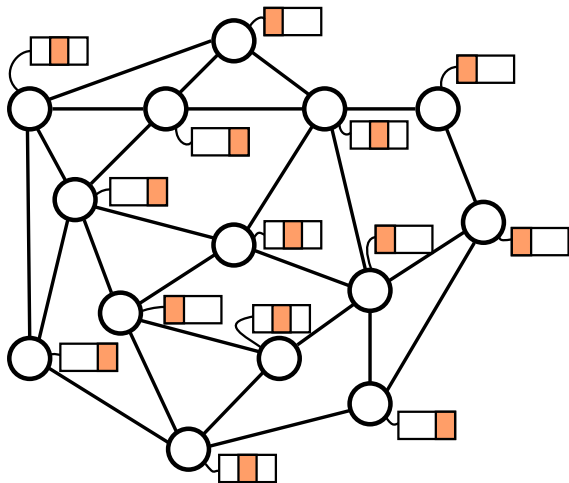
**Theorem :**

Uniform proofs can be scaled by a factor  $b(t)$ ,  
that is  $s_t(n) \in O(s_1(n)/b(t))$ .

# Spreading uniform proofs

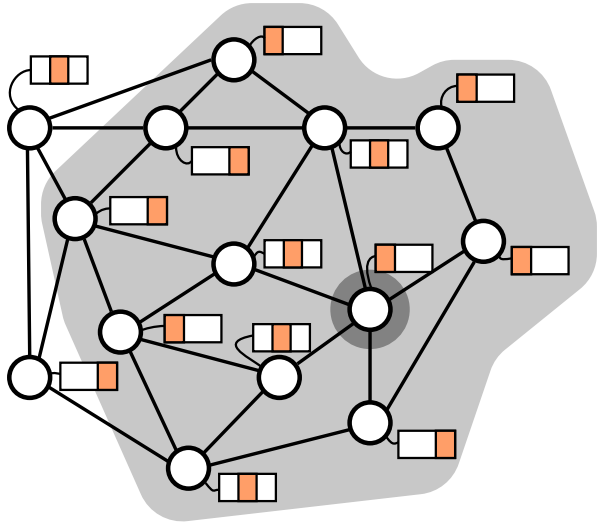


# Spreading uniform proofs

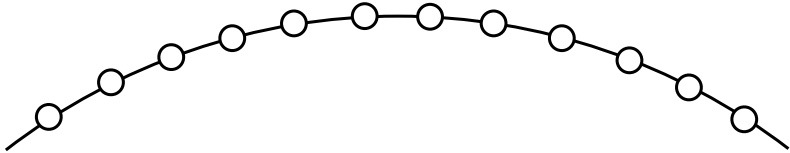




# Spreading uniform proofs



# Scaling on cycles



# Scaling on cycles

**Theorem :**

In cycles (and trees and grids), any proof can be scaled by a factor  $t$ , that is  $s_t(n) \in O(s_1(n)/t)$ .