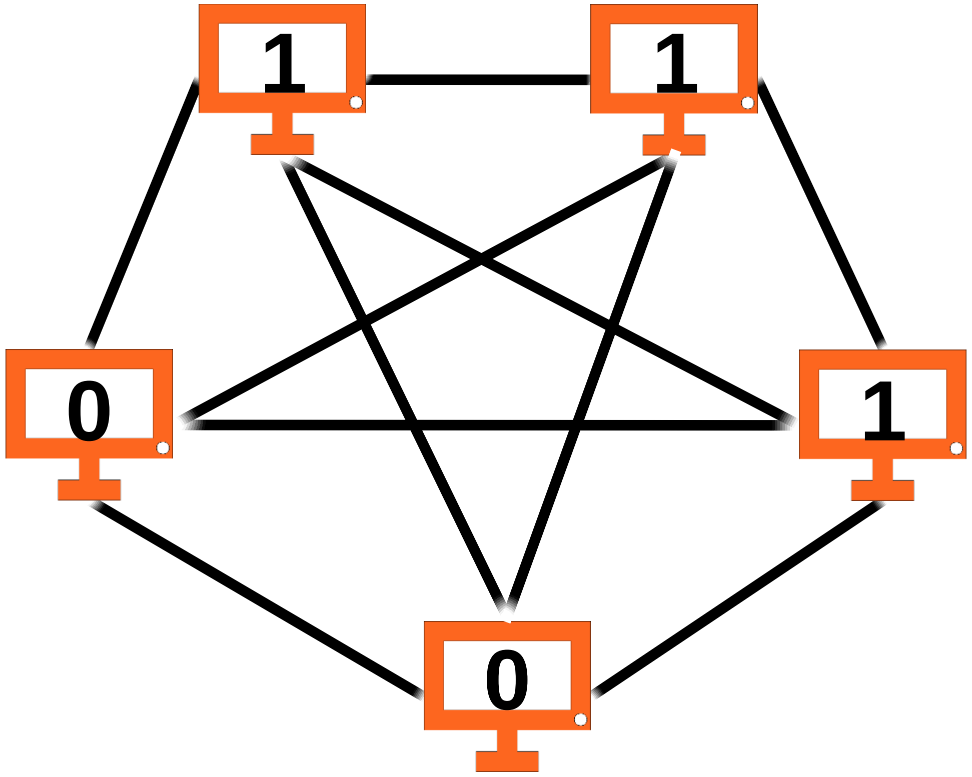
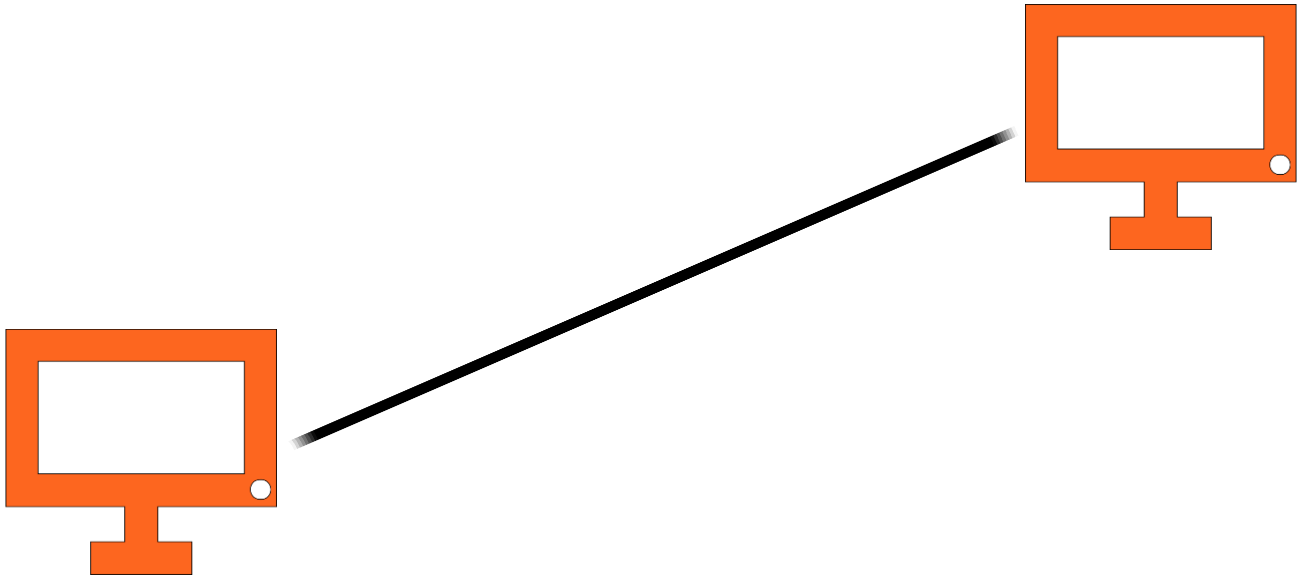
Consensus

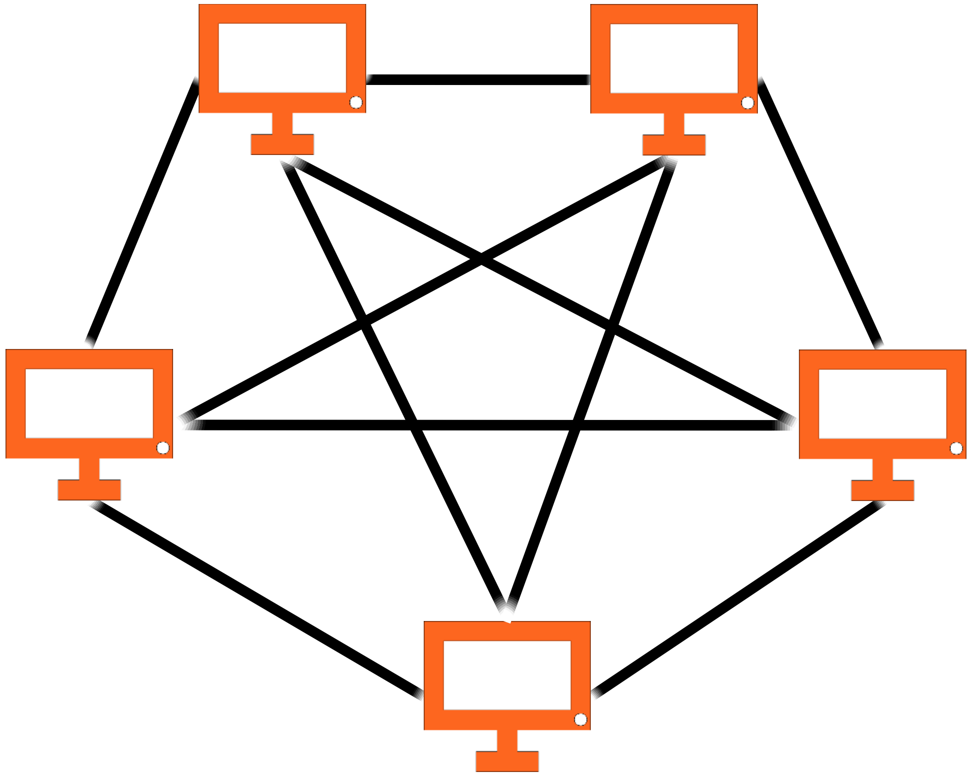


Consensus



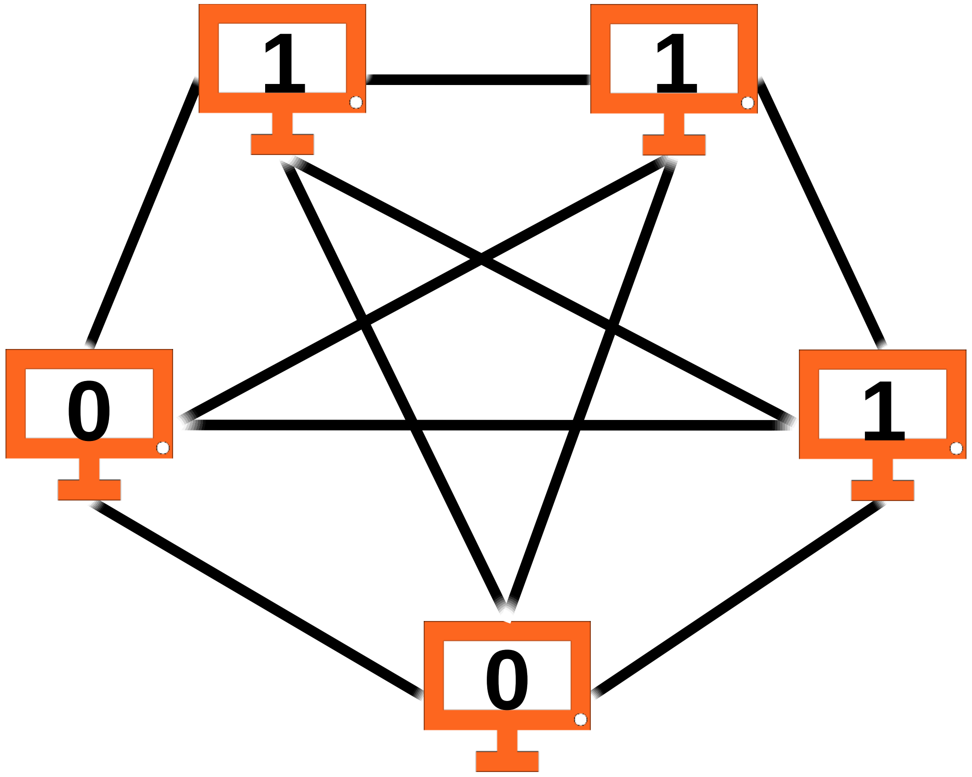
Assumptions in this lecture

- All-to-all communication



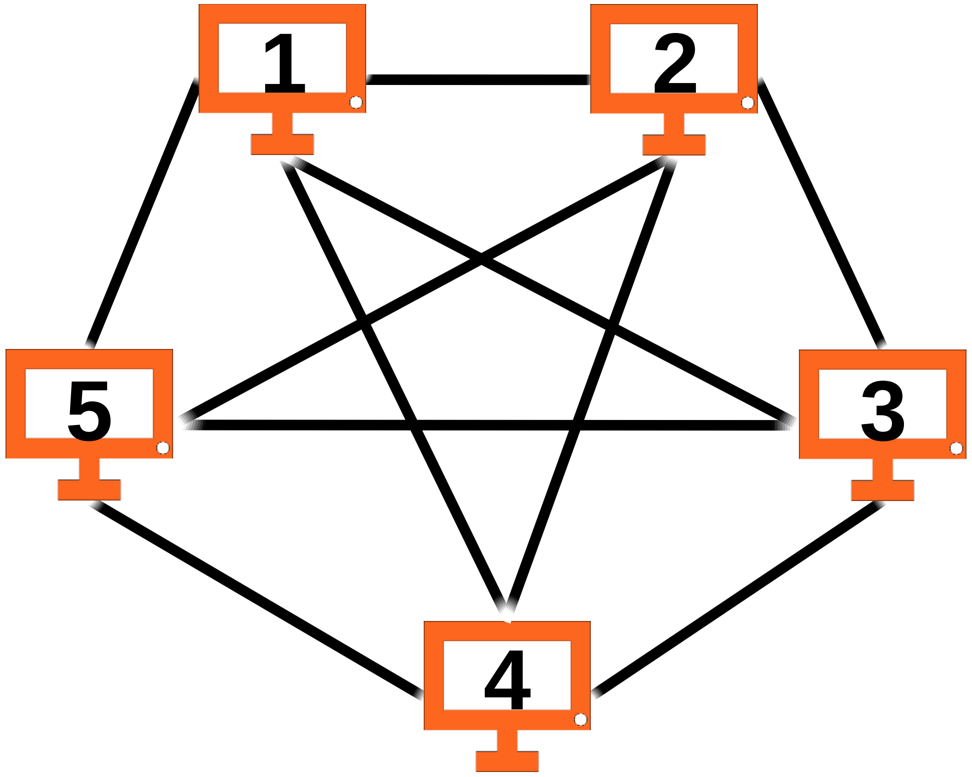
Assumptions in this lecture

- All-to-all communication
- Two possible input values, 0 and 1



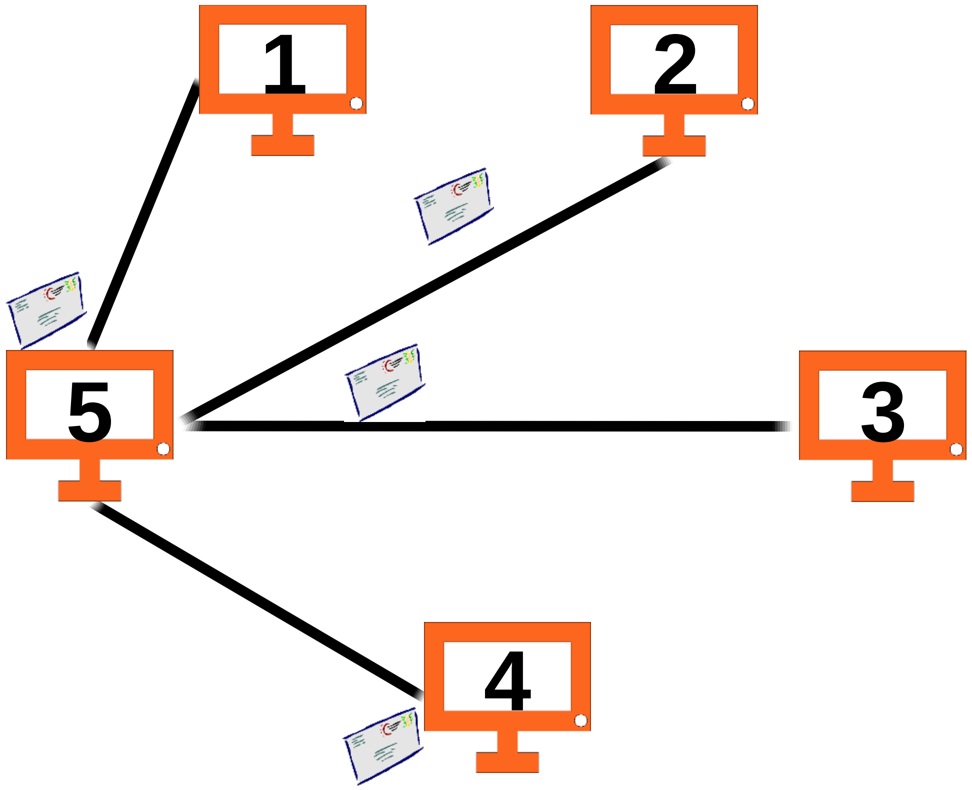
Assumptions in this lecture

- All-to-all communication
- Two possible input values, 0 and 1
- Unique IDs



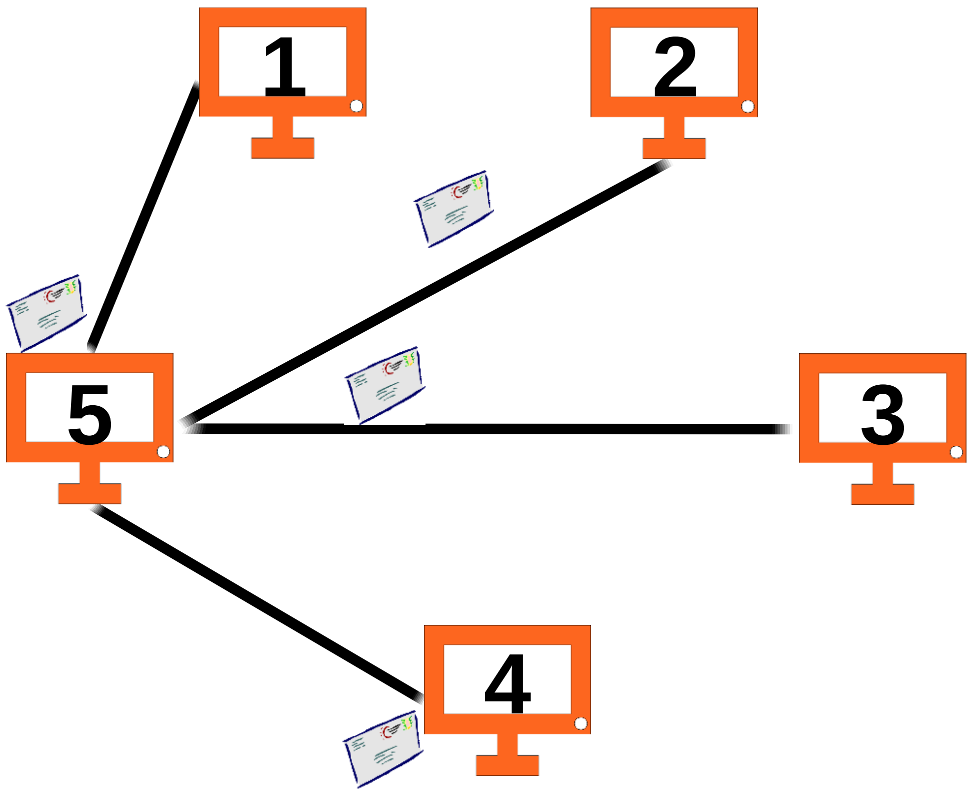
Assumptions in this lecture

- All-to-all communication
- Two possible input values, 0 and 1
- Unique IDs
- Asynchronous model

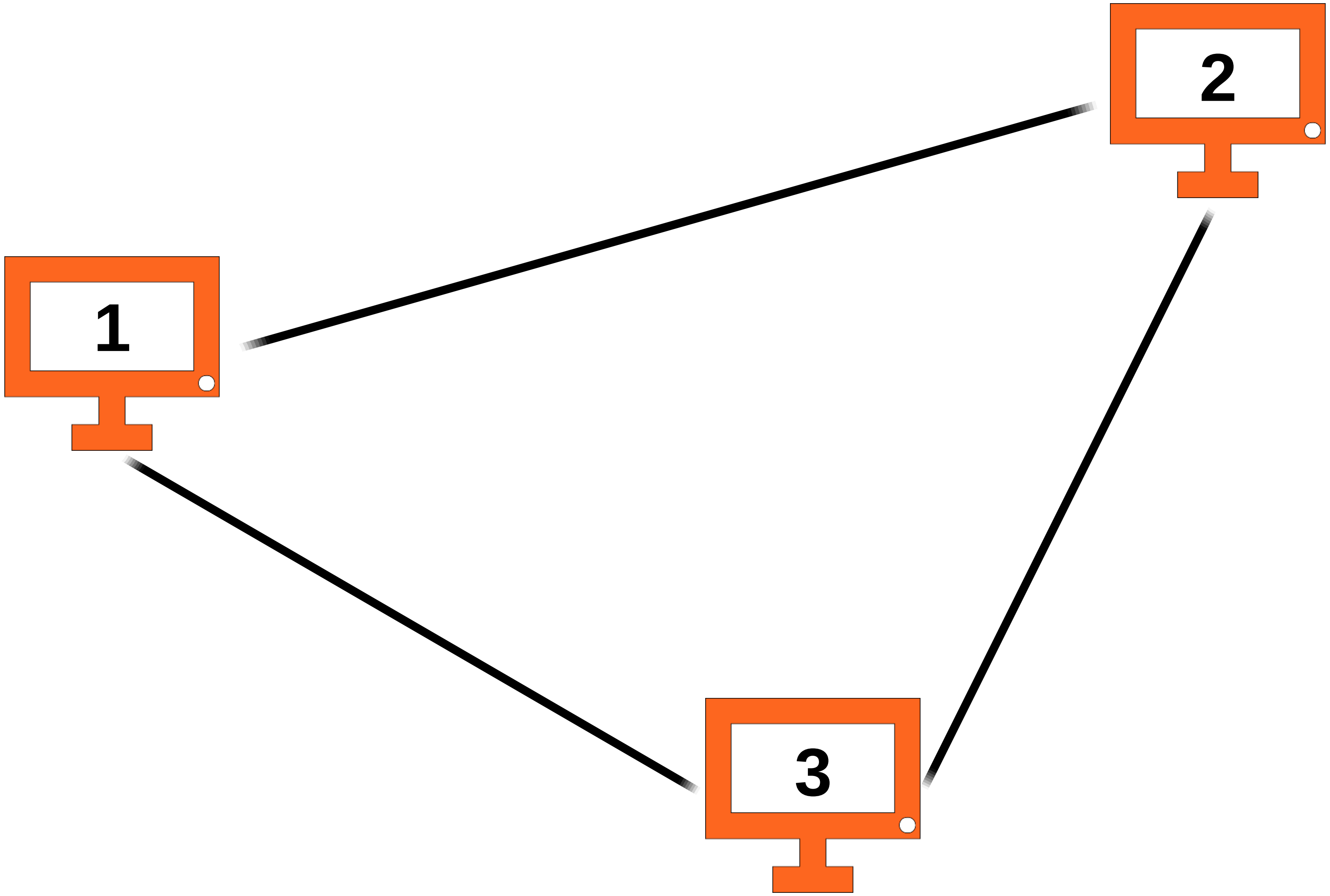


Recall: Asynchronous model

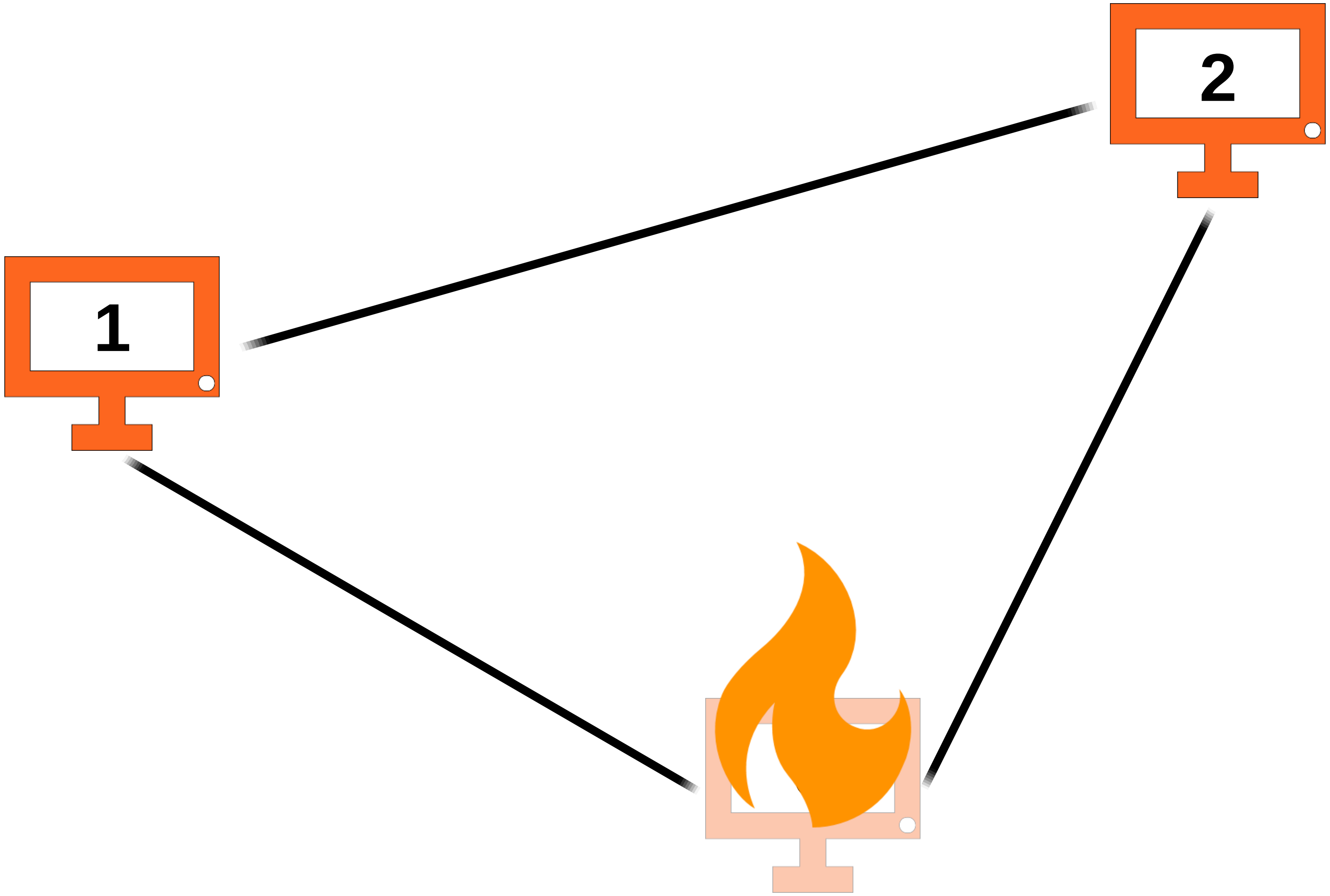
- Messages arrive eventually
  we cannot assume an upper bound on the message delay



Consensus



Consensus



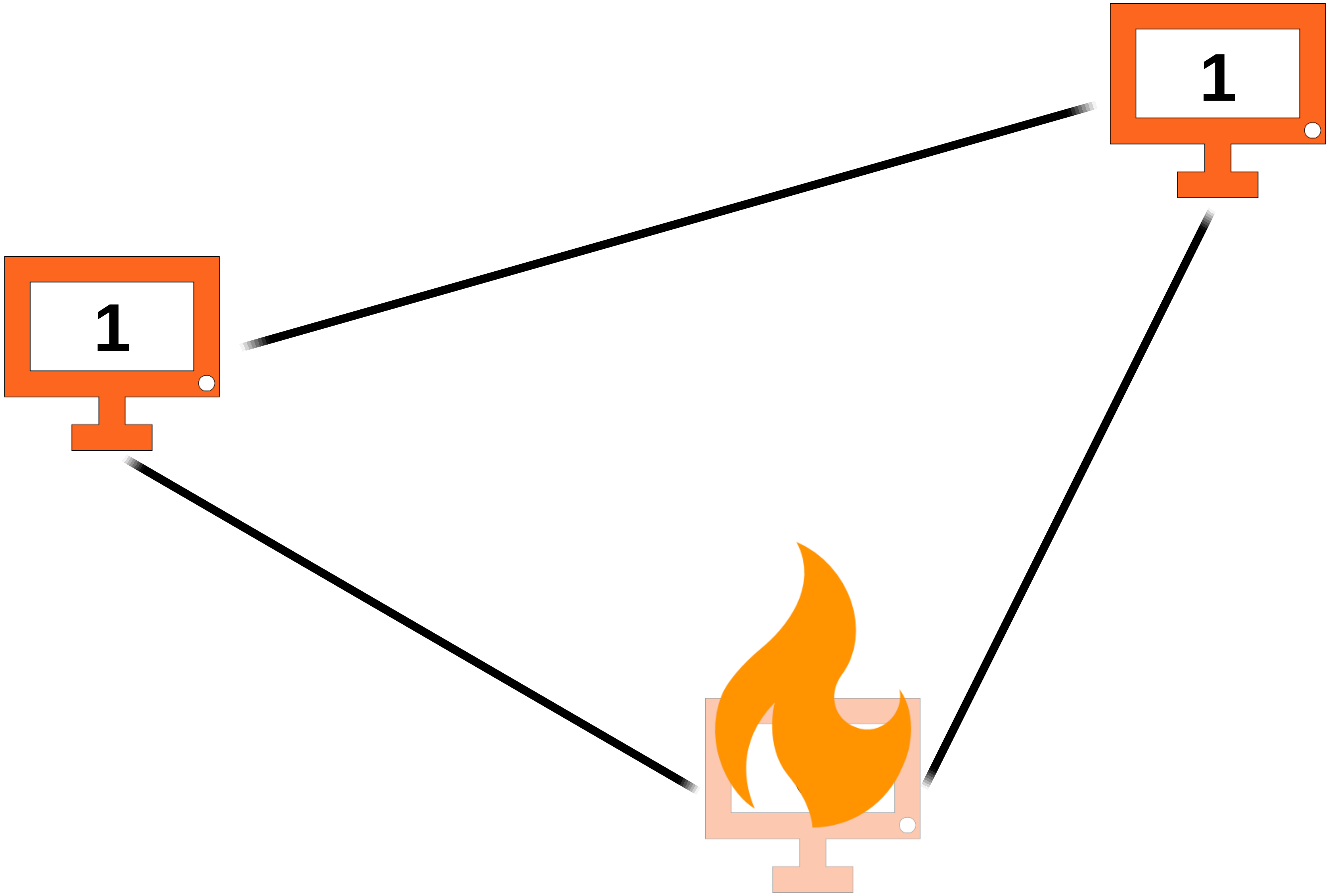
Crash failures

- A node can crash at any time,
- This node does not recover anymore,
- Messages that have been successfully sent by this node arrive eventually,
- We know how many nodes can crash – f of them

Requirements

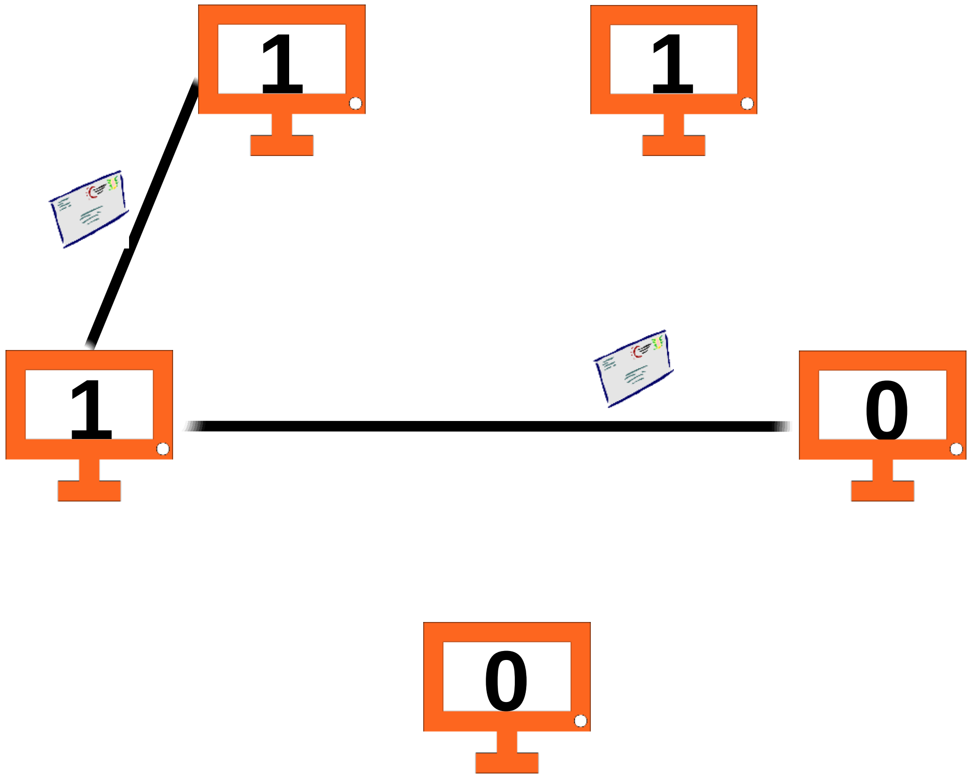
- *Agreement*
  the nodes agree on the same value
- *Termination*
  the nodes terminate in a finite time
- *Validity*
  the decision should be one of the inputs

Examples



No deterministic algorithm
in the asynchronous model
can solve consensus

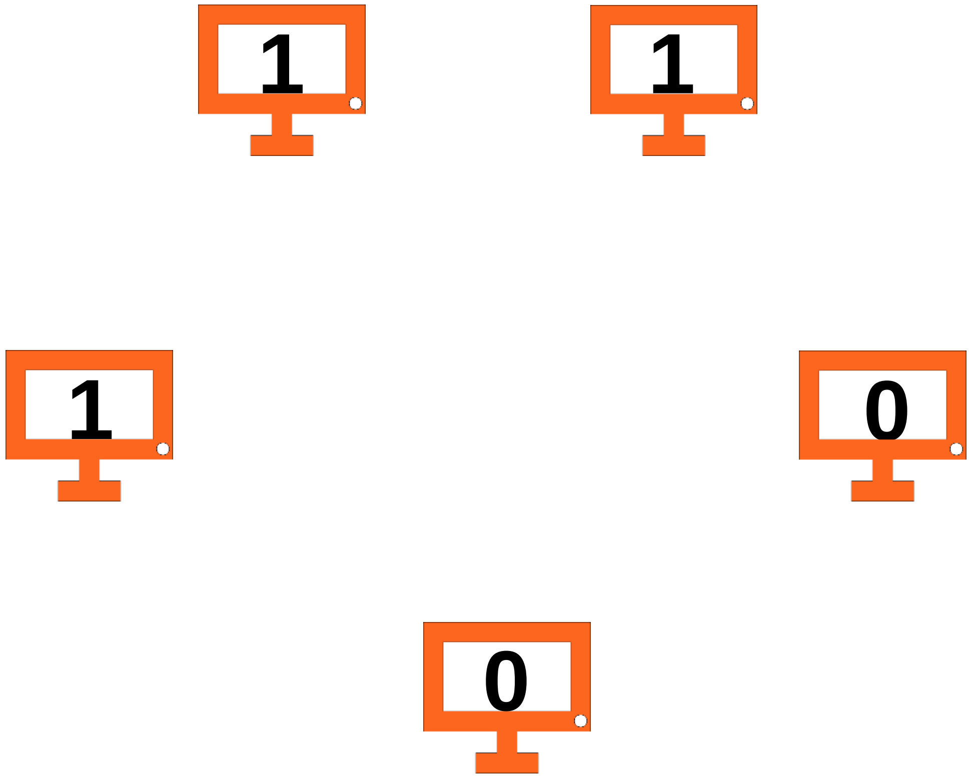
Algorithm Configuration



Fully described by:

- States
- Messages in transit

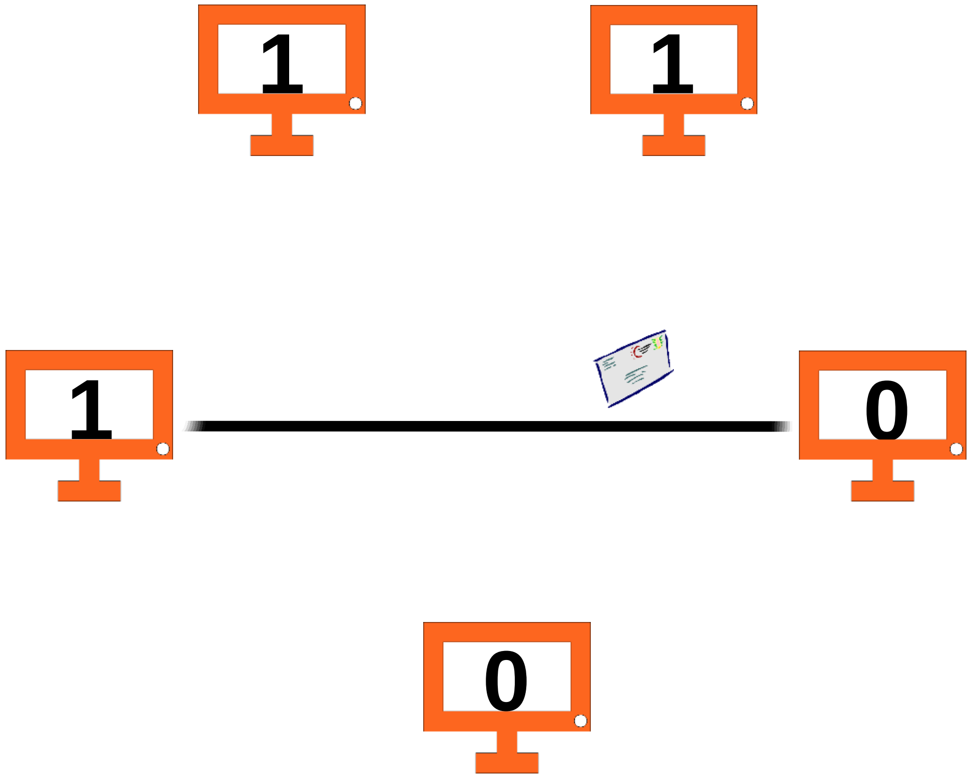
Initial Configuration C_0



Fully described by:

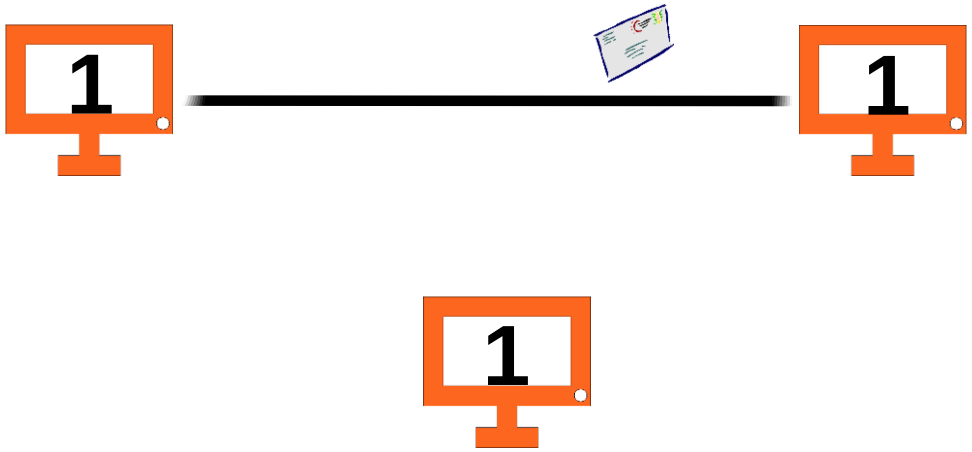
- IDs
- Inputs of the nodes

Bivalent Configuration B_i



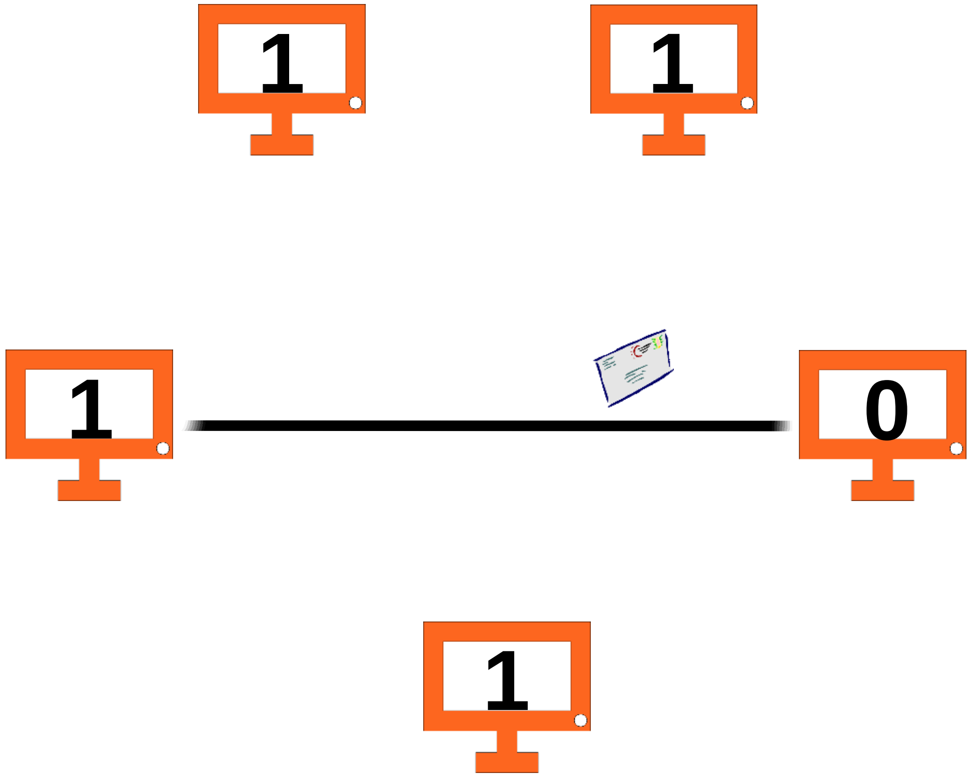
Configuration after which both decisions, 0 and 1, can follow

Univalent Configurations U_0 and U_1



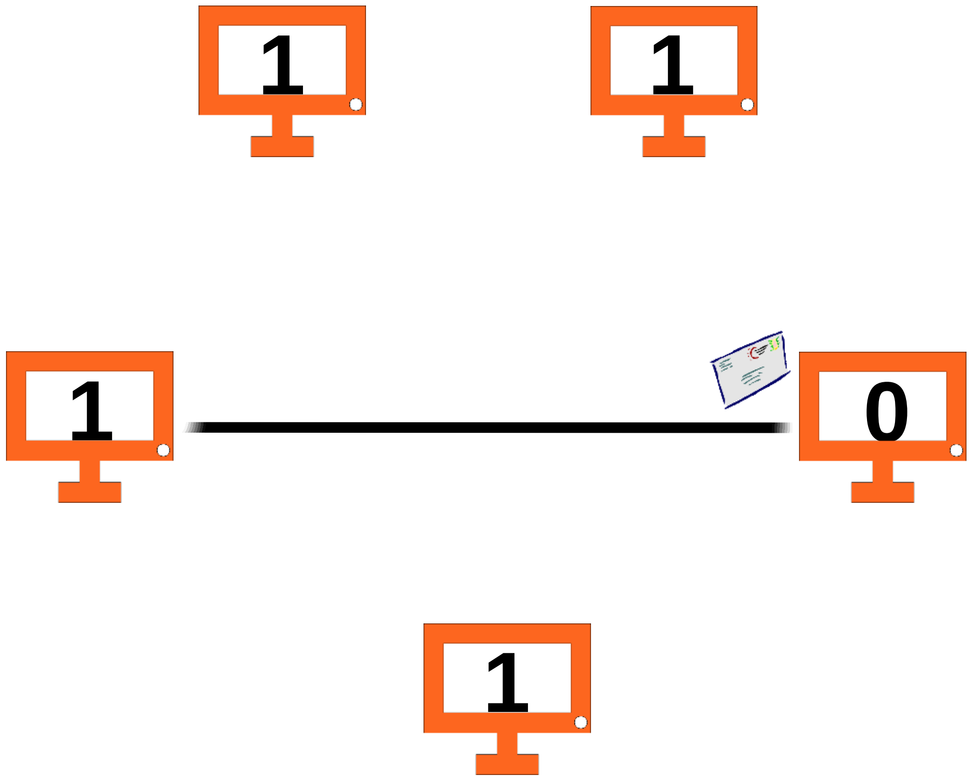
Configuration after which only one decision can follow, e.g., 1

Critical Configuration



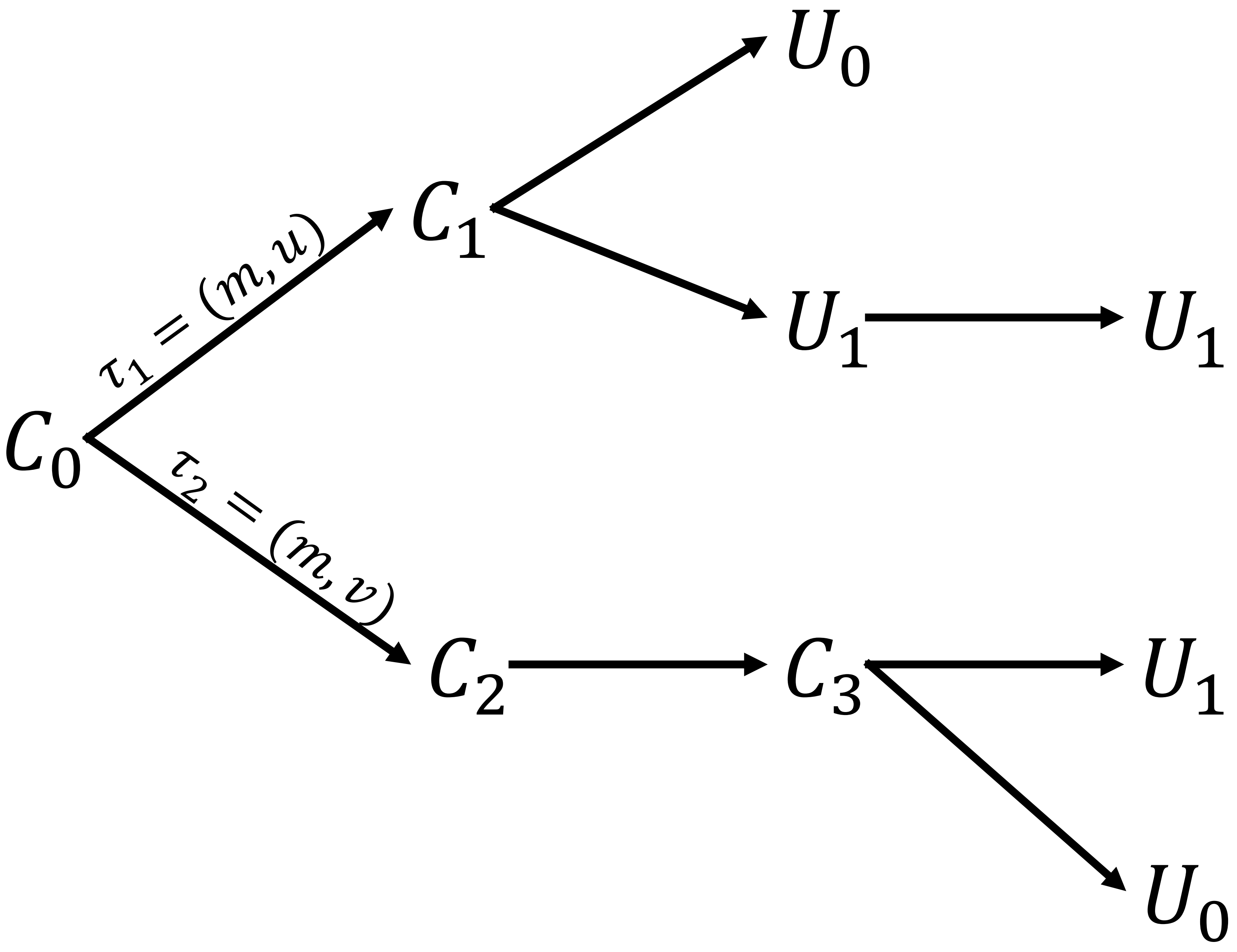
Last configuration after which both decisions can follow

Transition τ

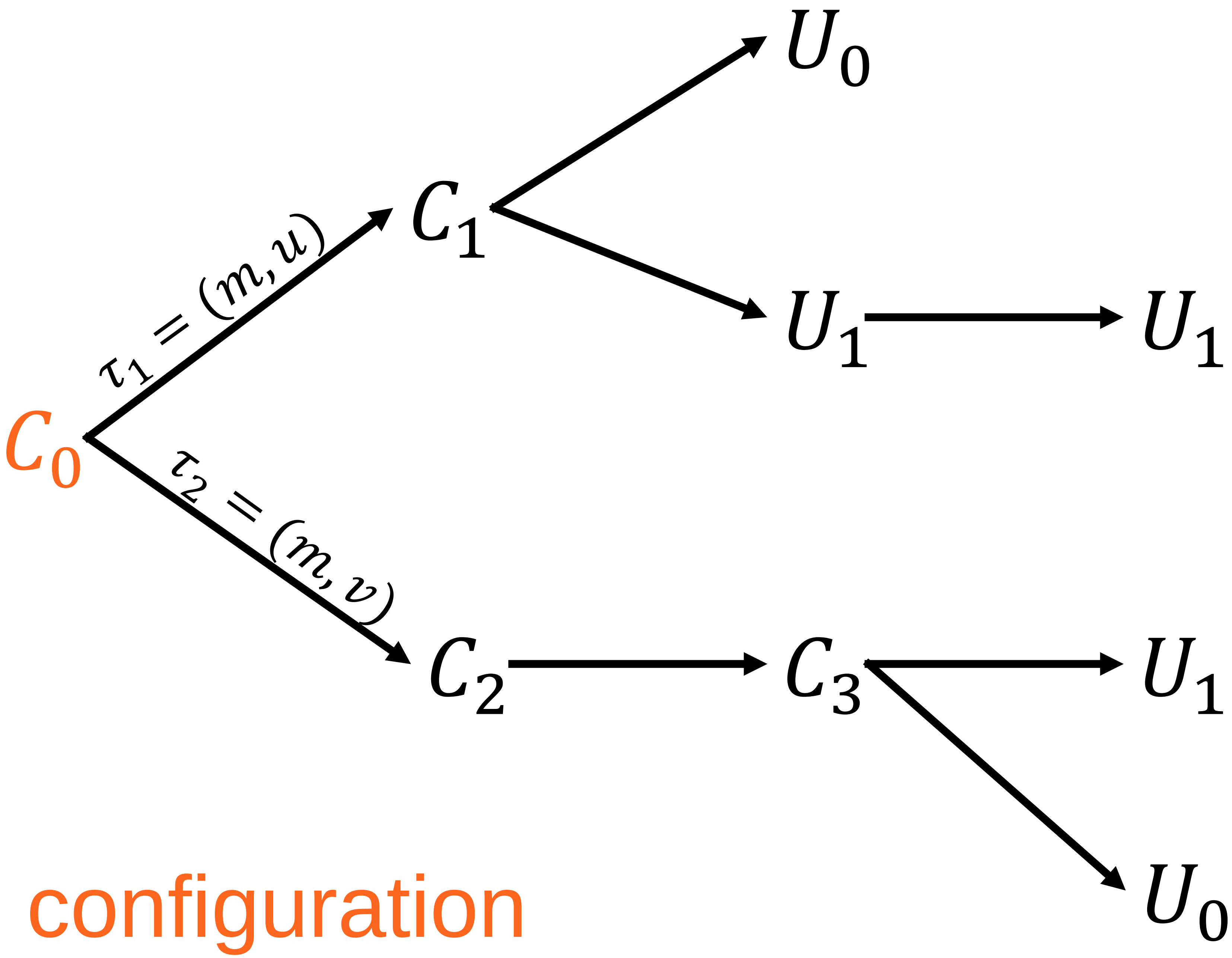


A transition $\tau = (m, u)$ is characterized by a node u receiving a message m

Configuration tree

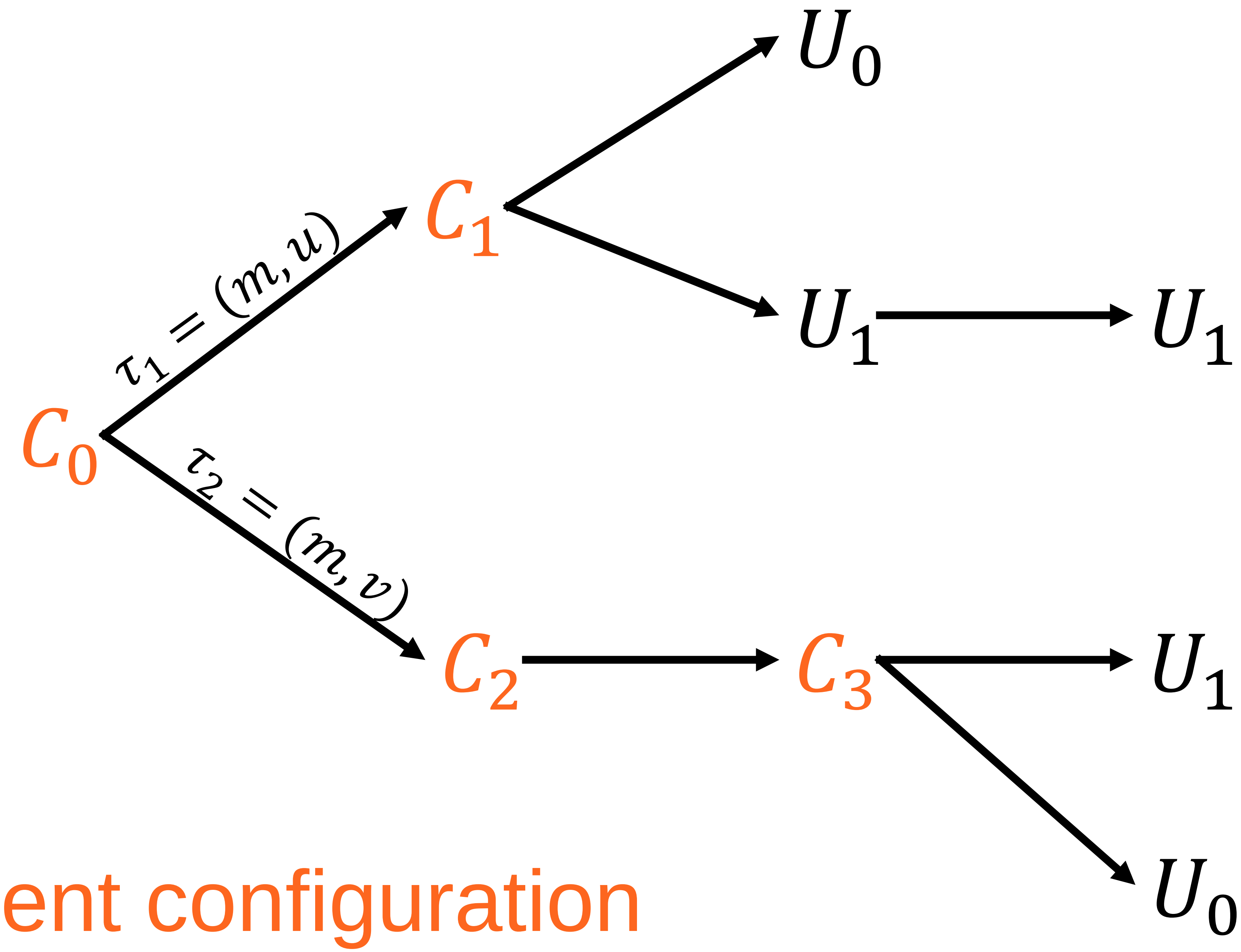


Configuration tree



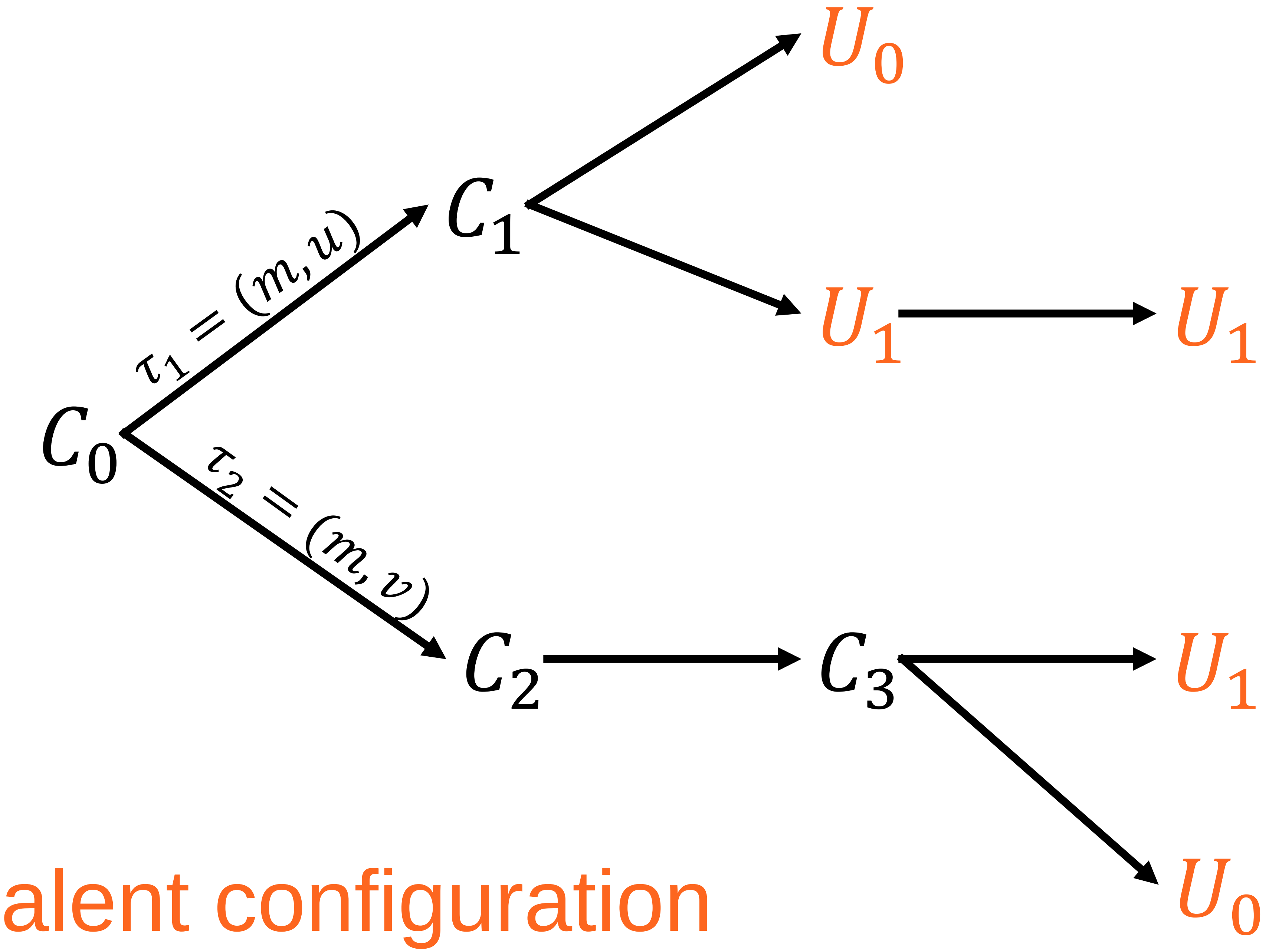
initial configuration

Configuration tree



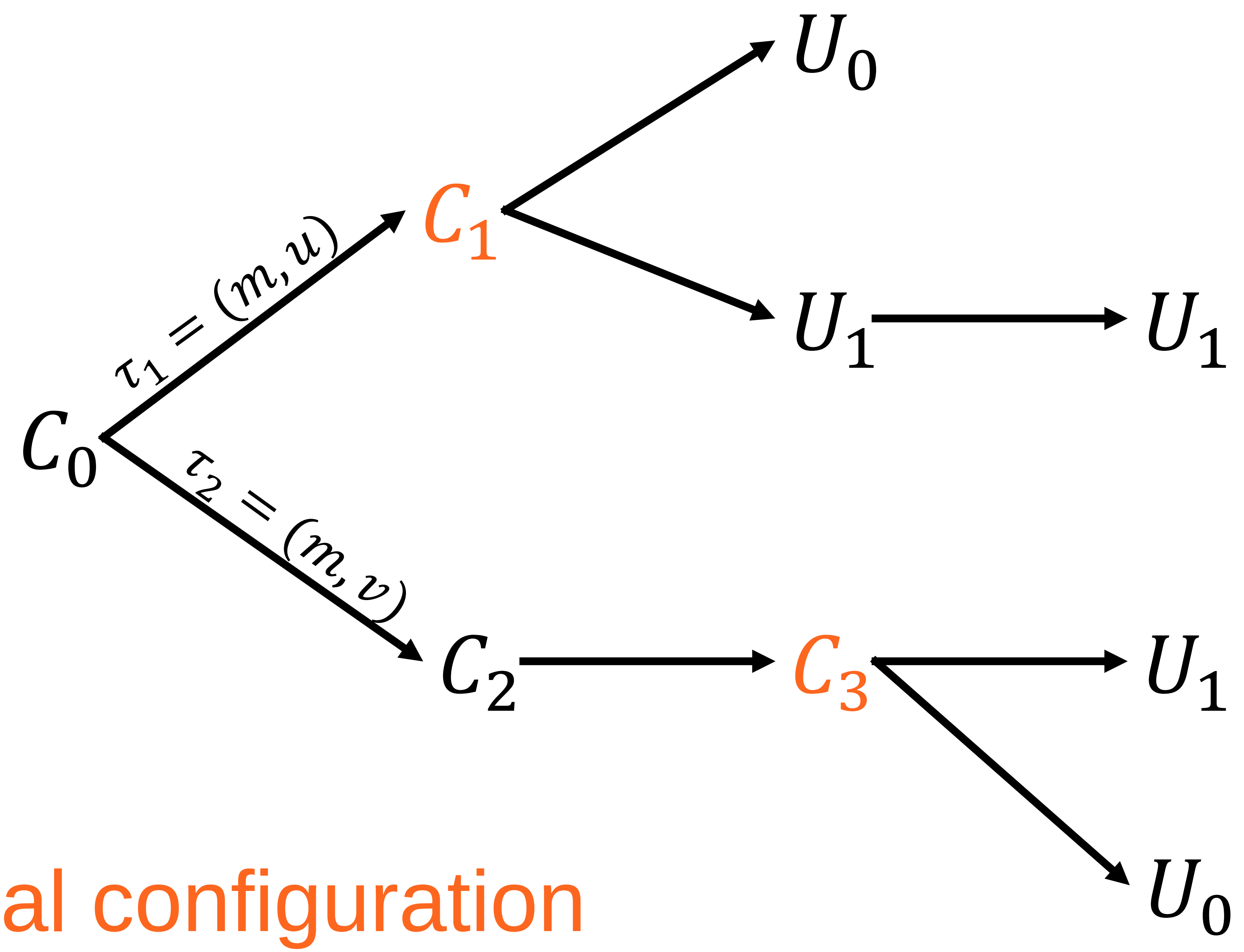
bivalent configuration

Configuration tree



univalent configuration

Configuration tree



critical configuration

Impossibility result - recipe

- There always exists a bivalent initial configuration

- There must exist a critical configuration

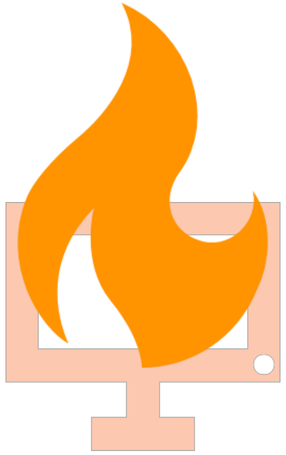
- The action of a single node decides whether the outcome is 0/1

Impossibility result - recipe

- There always exists a bivalent initial configuration

- There must exist a critical configuration

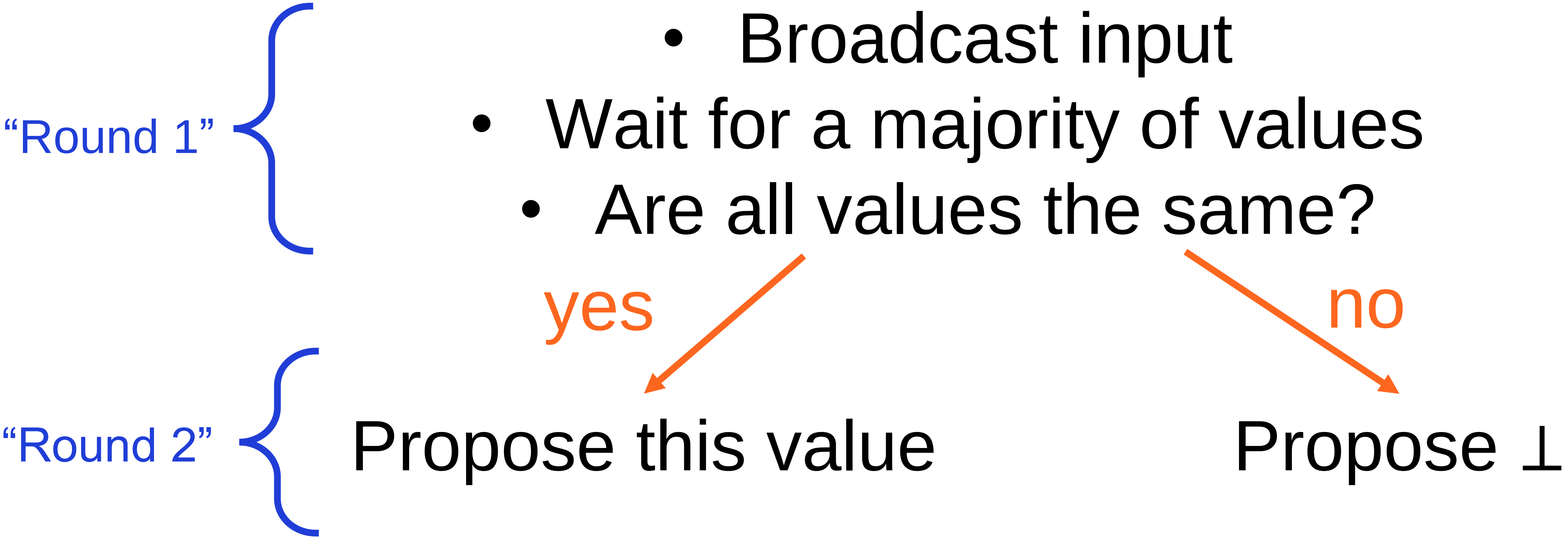
- The action of a single node decides whether the outcome is 0/1



Randomized Consensus, $\mathrm{f} < \frac{n}{2}$

“Round 1”

- Broadcast input
- Wait for a majority of values
- Are all values the same?



“Round 1”

- Broadcast input
- Wait for a majority of values
- Are all values the same?

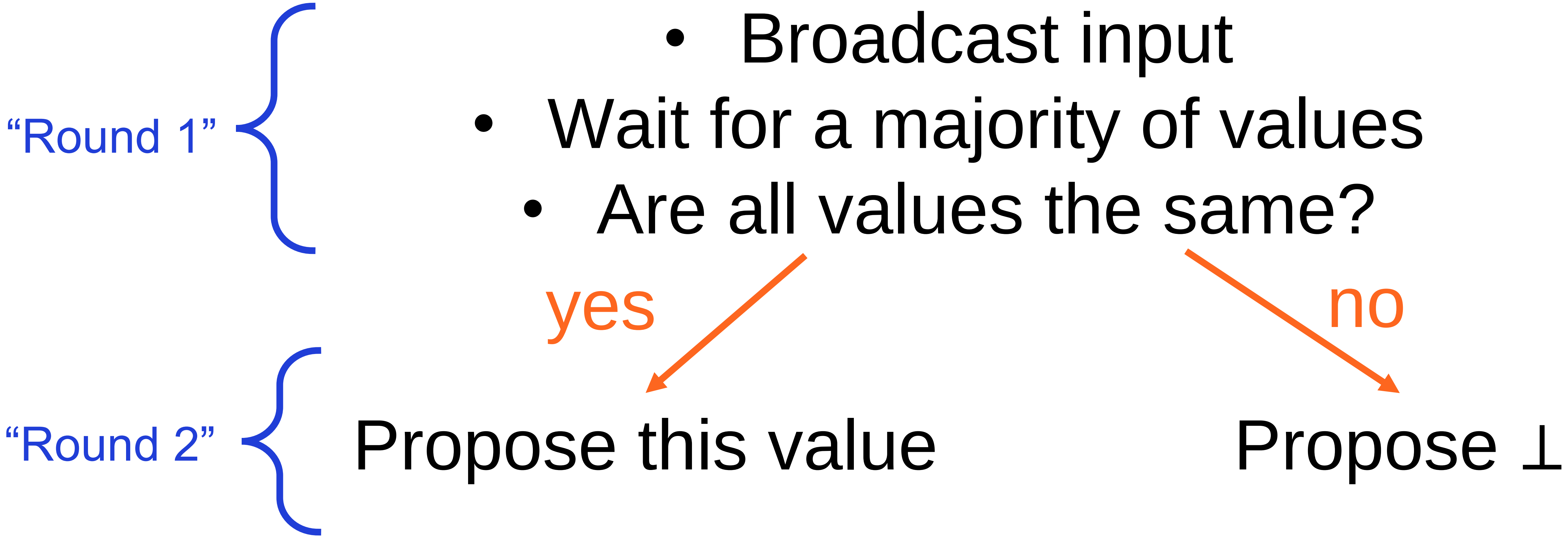
yes

no

“Round 2”

Propose this value

Propose ⊥



“Round 1”

- Broadcast input
- Wait for a majority of values
- Are all values the same?

yes

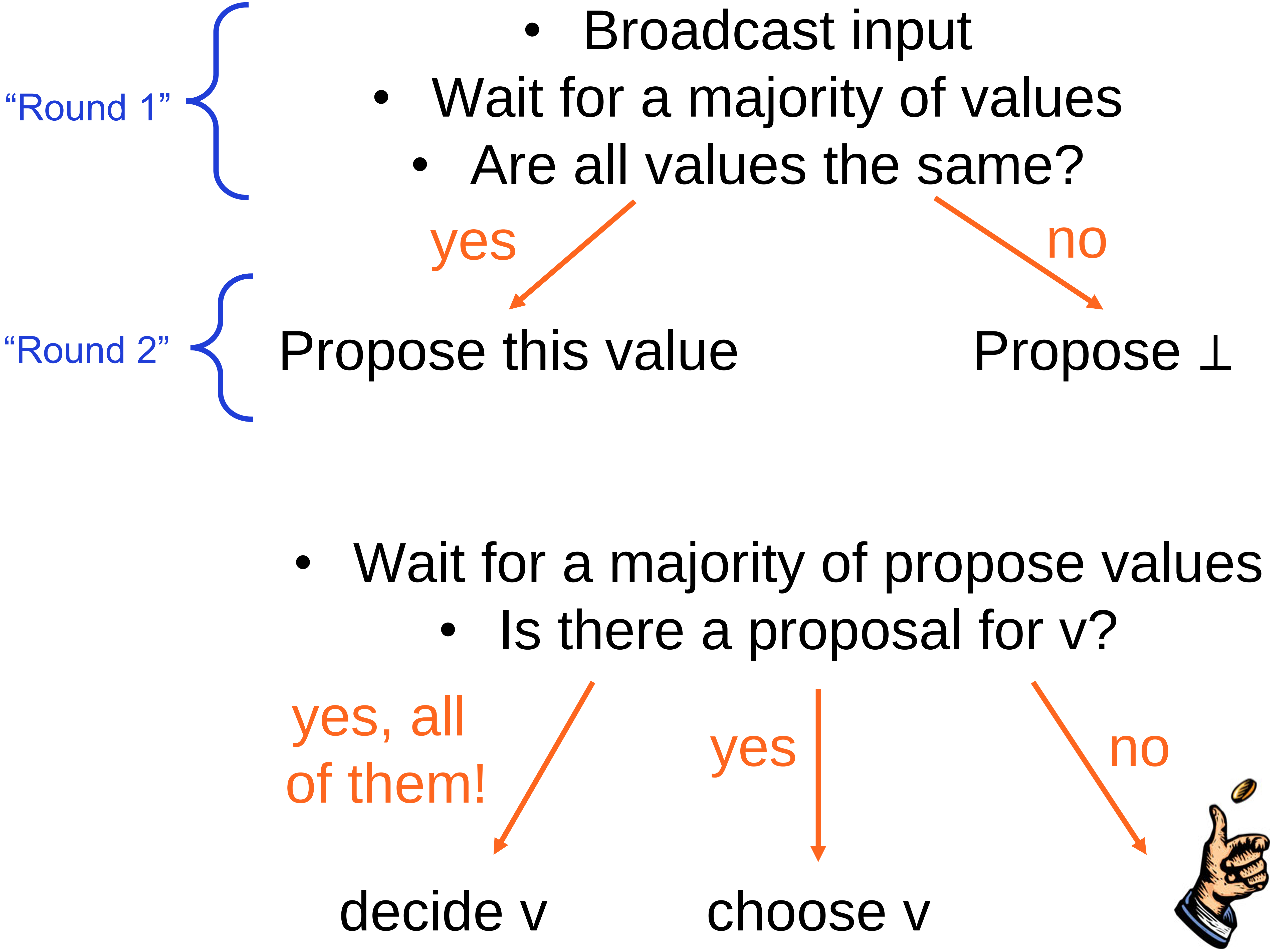
no

“Round 2”

Propose this value

Propose ⊥

- Wait for a majority of propose values
- Is there a proposal for v?



"Round 1"

- Broadcast input
- Wait for a majority of values
- Are all values the same?

yes

no

"Round 2"

Propose this value

Propose ⊥

- Wait for a majority of propose values
- Is there a proposal for v?

yes, all of them!

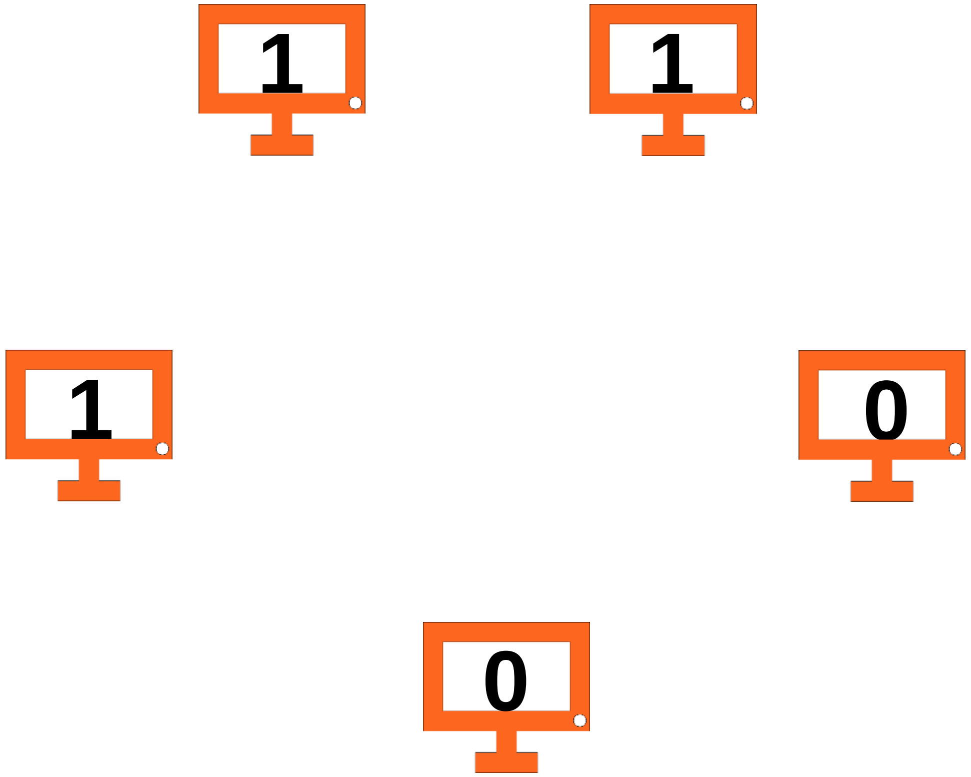
yes

no

decide v

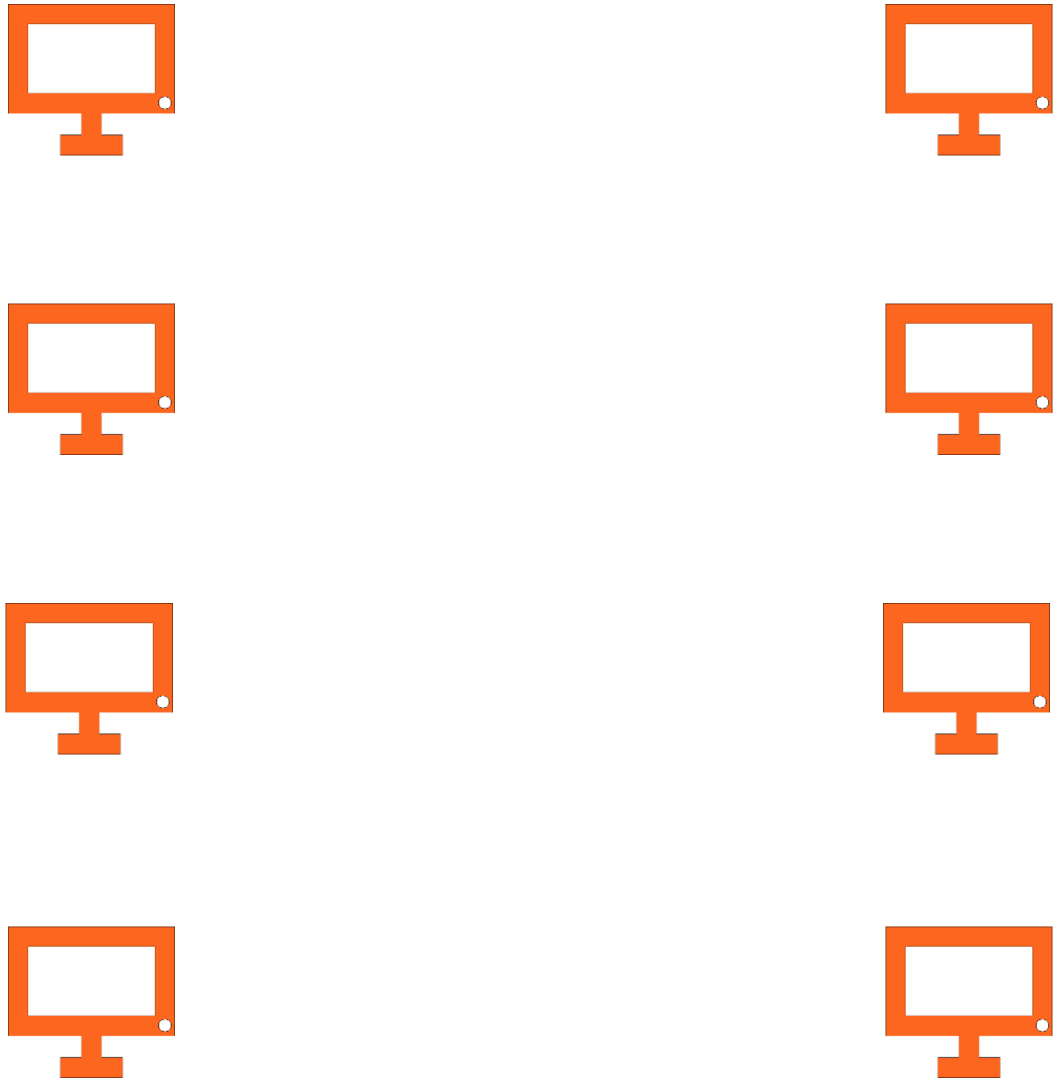
choose v

Example



Can any algorithm handle $\mathrm{f} = \frac{n}{2}$ crashes?

Example



Improving the coinflip, $\mathrm{f} < \frac{n}{3}$

Requirement:
output 0 and 1 with constant probability

- Biased local coin:
    - 0 with probability $1/n$
    - 1 with probability $1 - \frac{1}{n}$
- Broadcast coinflip

- Biased local coin:
  - 0 with probability $1/n$
  - 1 with probability $1-\frac{1}{n}$
- Broadcast coinflip

- Wait for $n-f$ coins, store them in C_u
- Broadcast C_u

- Biased local coin:
    - 0 with probability $1/n$
    - 1 with probability $1 - \frac{1}{n}$
- Broadcast coinflip

- Wait for $n - f$ coins, store them in C_u
- Broadcast C_u

- Wait for $n - f$ coin sets
- Does one of them contain 0?

- Biased local coin:
    - ➢ 0 with probability $1/n$
    - ➢ 1 with probability $1 - \frac{1}{n}$
- Broadcast coinflip

- Wait for $n - f$ coins, store them in C_u
- Broadcast C_u

- Wait for $n - f$ coin sets
- Does one of them contain 0?

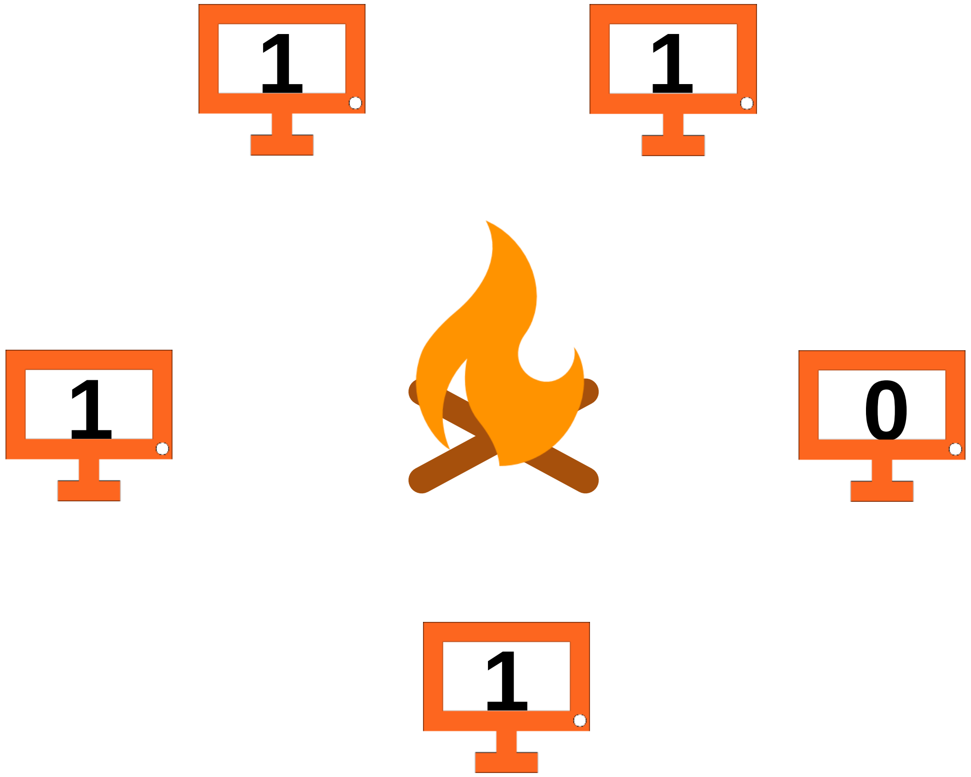
yes

no

decide 0

decide 1

Analysis



Learning goals

- **Problems:** Consensus
- **Distributed models:** asynchronous all-to-all communication
- **Impossibility results:**
  - Impossibility of deterministic asynchronous consensus
  - Impossibility of consensus with $n/2$ failures
- **Algorithms:**
  - Randomized consensus algorithm with $f < n/2$
  - Biased local coin