

Laboratorio di Reti e Sistemi Distribuiti

17: Leader Election

Roberto Marino, PhD¹
`roberto.marino@unime.it`

¹Dipartimento di Matematica, Informatica, Fisica e Scienze della Terra
Future Computing Research Laboratory
Università di Messina

Last Update: 15th May 2025

Il problema dell'elezione del Leader

Definizione del problema

Si supponga che tutti i processori di una rete abbiano una variabile di stato, ad esempio `leader`, inizialmente impostata su `"unknown"`. Diciamo che un leader viene eletto quando uno e un solo processore ha la variabile di stato impostata su `true` e tutti gli altri la impostano su `false`. Eleggere un leader.

Algoritmo Floodmax

Descrizione Informale

In ogni round di comunicazione, ogni agente invia ai suoi vicini l'UID massimo che ha ricevuto fino a quel momento. Questa operazione viene ripetuta per $diam(S)$ round. All'ultimo round, ogni agente confronta l'UID massimo ricevuto con il proprio e si dichiara leader se coincidono, altrimenti non leader.

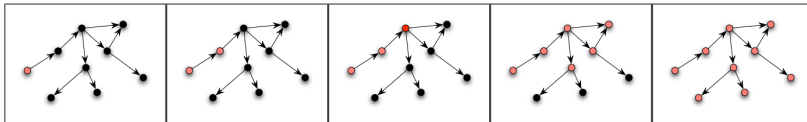


Figure 1.16 Execution of the FLOODMAX ALGORITHM. The diameter of the network is 4. In the leftmost frame, the agent with the maximum UID is colored in red. After four communication rounds, its message has been received by all agents.

Alphabet: $\mathbb{A} = \{1, \dots, n\} \cup \{\text{null}\}$

Processor State: $w = (\text{my-id}, \text{max-id}, \text{leader}, \text{round})$, where

$\text{my-id} \in \{1, \dots, n\},$	initially: $\text{my-id}^{[i]} = i$ for all i
$\text{max-id} \in \{1, \dots, n\},$	initially: $\text{max-id}^{[i]} = i$ for all i
$\text{leader} \in \{\text{false}, \text{true}, \text{unknown}\},$	initially: $\text{leader}^{[i]} = \text{unknown}$ for all i
$\text{round} \in \{0, 1, \dots, \text{diam}(S)\},$	initially: $\text{round}^{[i]} = 0$ for all i

function $\text{msg}(w, i)$

```
1: if round < diam(S) then
2:   return max-id
3: else
4:   return null
```

function $\text{stf}(w, y)$

```
1: new-id := max{max-id, largest identifier in y}
2: case
3:   round < diam(S):   new-lead := unknown
4:   round = diam(S) AND max-id = my-id:   new-lead := true
5:   round = diam(S) AND max-id > my-id:   new-lead := false
6: return (my-id, new-id, new-lead, round + 1)
```

FLOODMAX: Analisi di Complessità

Complessità temporale

Uguale a $diam(S)$

Complessità di comunicazione

Complessità in $O(diam(S)dim(E_{cmm}))$

Definizione

Il diametro di un grafo è la distanza massima fra i cammini minimi tra due nodi

Algoritmo LCR (Lann–Chang–Roberts)

Descrizione Informale

Dato un grafo orientato ad anello, in ogni round di comunicazione, ogni agente invia ai suoi vicini l'UID massimo che ha ricevuto fino a quel momento (gli agenti non registrano il numero di cicli di comunicazione). Quando l'agente con il massimo UID riceve il suo proprio UID da un vicino, si dichiara leader.

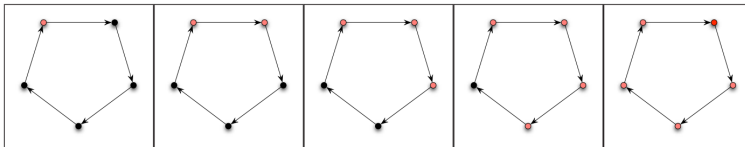


Figure 1.17 Execution of the LCR ALGORITHM. In the leftmost frame, the agent with the maximum UID is colored in red. After five communication rounds, this agent receives its own UID from its in-neighbor and declares itself the leader.

Synchronous Network: ring digraph

Distributed Algorithm: LCR

Alphabet: $\mathbb{A} = \{1, \dots, n\} \cup \{\text{null}\}$

Processor State: $w = (\text{my-id}, \text{max-id}, \text{leader}, \text{snd-flag})$, where

$\text{my-id} \in \{1, \dots, n\},$	initially: $\text{my-id}^{[i]} = i$ for all i
$\text{max-id} \in \{1, \dots, n\},$	initially: $\text{max-id}^{[i]} = i$ for all i
$\text{leader} \in \{\text{true}, \text{false}, \text{unknown}\},$	initially: $\text{leader}^{[i]} = \text{unknown}$ for all i
$\text{snd-flag} \in \{\text{true}, \text{false}\},$	initially: $\text{snd-flag}^{[i]} = \text{true}$ for all i

function $\text{msg}(w, i)$

```
1: if  $\text{snd-flag} = \text{true}$  then
2:   return  $\text{max-id}$ 
3: else
4:   return  $\text{null}$ 
```

function $\text{stf}(w, y)$

```
1: case
2:   ( $y$  contains only  $\text{null}$  msgs) OR (largest identifier in  $y < \text{my-id}$ ):
3:      $\text{new-id} := \text{max-id}$ 
4:      $\text{new-lead} := \text{leader}$ 
5:      $\text{new-snd-flag} := \text{false}$ 
6:   (largest identifier in  $y = \text{my-id}$ ):
7:      $\text{new-id} := \text{max-id}$ 
8:      $\text{new-lead} := \text{true}$ 
9:      $\text{new-snd-flag} := \text{false}$ 
10:  (largest identifier in  $y > \text{my-id}$ ):
11:     $\text{new-id} := \text{largest identifier in } y$ 
12:     $\text{new-lead} := \text{false}$ 
13:     $\text{new-snd-flag} := \text{true}$ 
14:  return ( $\text{my-id}, \text{new-id}, \text{new-lead}, \text{new-snd-flag}$ )
```

LCR: Analisi di Complessità

Complessità temporale

Uguale al numero dei nodi n

Complessità di comunicazione

Complessità in $\Theta(n^2)$