

## Laborator nr. 12

### Alegerea liderului (2)

#### 1 Algoritmul FloodMax Sincron

Algoritmul FloodMax sincron este un algoritm de alegere a liderului destinat topologiilor de tip graf.

1. Graful suport  $G = (V, E)$  este un graf conex. Fiecare unitate de procesare cunoaște diametrul  $d$  al grafului.
2. Fiecare unitate de procesare conține o înregistrare a identificadorului maxim cunoscut. Inițial, această înregistrare este propriul identificador.
3. În cadrul fiecărei runde, fiecare unitate de procesare trimite maximul cunoscut tuturor unităților vecine din graful suport.
4. După  $d$  runde, dacă maximul cunoscut este propriul identificador, atunci unitatea de procesare se declară *lider*, altfel se consideră *non – lider*.

Algoritmul 1 expune pseudocodul algoritmului FloodMax sincron.

- *Notatii:*

- $id_i$  – identificadorul unității de procesare  $p_i$ ;
- $R_i$  – colecția de mesaje recepționate de unitatea de procesare  $p_i$  în cadrul unei runde;
- $max\_id_i$  – stochează cel mai mare identificador cunoscut de  $p_i$ ;
- $statut_i$  – starea finală a unității de procesare  $p_i$  (*lider* sau *non – lider*).

- *Premise:*

- Inițial  $max\_id_i = id_i$  și  $statut_i = necunoscut$ .

#### 2 Aplicații

Implementați, utilizând MPI, algoritmul prezentat în laborator. Topologia suport este prezentată în Figura 1.

---

**Algoritmul 1** Algoritmul FloodMax sincron

---

```
1: function FLOODMAX_LE( $G, d, p_i, max\_id_i, statut_i$ )
2:   for  $runda \leftarrow 1$  to  $d$  do
3:     for each  $m \in R_i$  do
4:       if  $max\_id_i < m$  then
5:          $max\_id_i \leftarrow m$ 
6:       end if
7:     end for
8:     trimite  $max\_id_i$  tuturor vecinilor
9:   end for
10:  if  $max\_id_i = id_i$  then
11:     $statut_i \leftarrow lider$ 
12:  else
13:     $statut_i \leftarrow non - lider$ 
14:  end if
15: end function
```

---

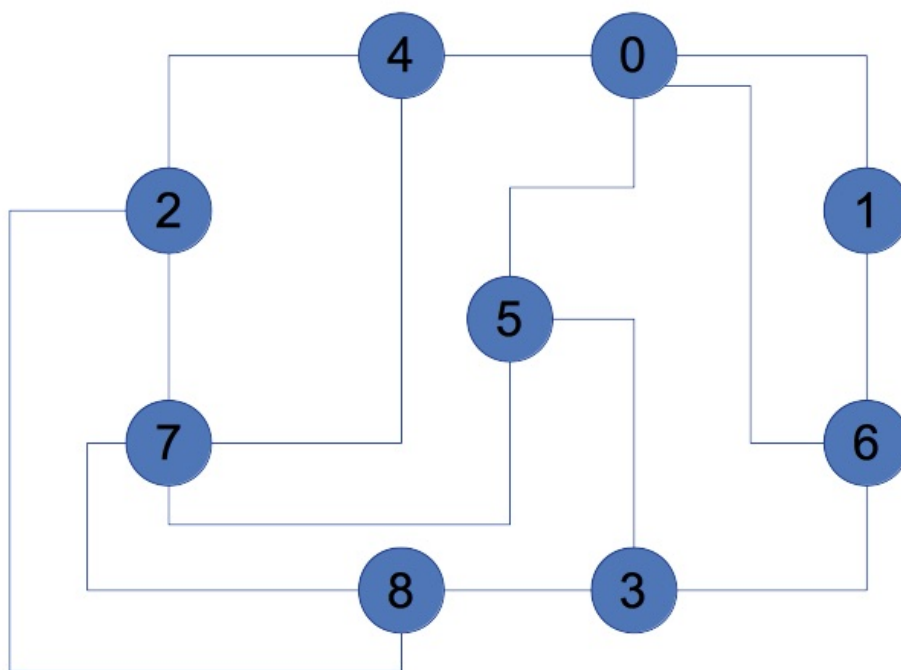


Figura 1: Graful suport pentru implementarea Algoritmului 1