

# Il problema ELECTION

"rompere la simmetria"

Scopo: Individuare una entità specifica  
Tra tante entità autonome e OMOGENEE  
Tale entità sarà chiamata LEADER e  
le altre FOLLOWER

Applicazioni: Per certi lavori serve avere una  
unità centrale che diventi coordinatrice  
per le altre entità

## - Risultato di impossibilità:

Tale: È impossibile deterministicamente individuare un LEADER sotto le restrizioni R

Idea della prova: Siano  $x, y \in E$  ovviamente omogenee. Esse sono nello stesso stato e inizializzate nello stesso modo. Eseguono entrambe lo stesso algoritmo e si ritrovano ancora in perfetta simmetria

## - Risultato di possibilità (ovvio):

Sotto le restrizioni RI, la starting entità diventa immediatamente LEADER ...

"Il problema però è risolto dall'esterno e non dal sistema!"

## Nuova Restrizione: Initial Distinct Values (ID)

Notazioni:  $R \cup \{ID\} = IR$ ,  $id(x) = \text{nome di } x \in E$   
o valore di  $x$

### Strategie di soluzione:

#### a) Elect Minimum

- 1) Trova  $id(x)$  minimo e fai  $x$  LEADER
- 2)  $\forall y \neq x \in E$   $y$  diventa FOLLOWER

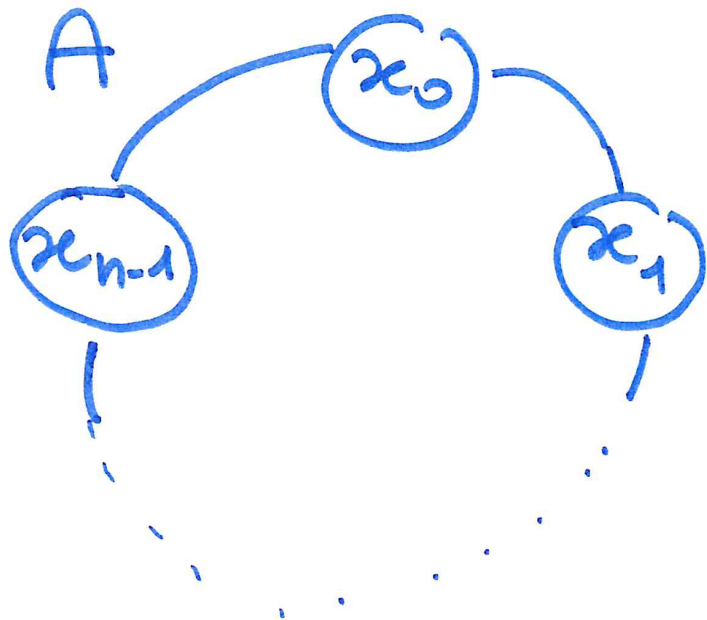
#### b) Elect Minimum Initiator

- 1) Trova  $id(x)$  minimo tra le sole entità initiator  
ed eleggi  $x$  LEADER
- 2)  $\forall y \neq x \in E$ ,  $y$  diventa FOLLOWER

Risolviamo il problema 2) in una Topologia particolare: RING = ANELLO

Topologia Ring: le entità sono disposte su un anello

$$A = (x_0, x_1, \dots, x_{n-1}) \quad \text{caso } m=n$$



Restrizioni:

$IR \cup \{ \text{conoscenza di } x \in A \}$   
di essere in un RING

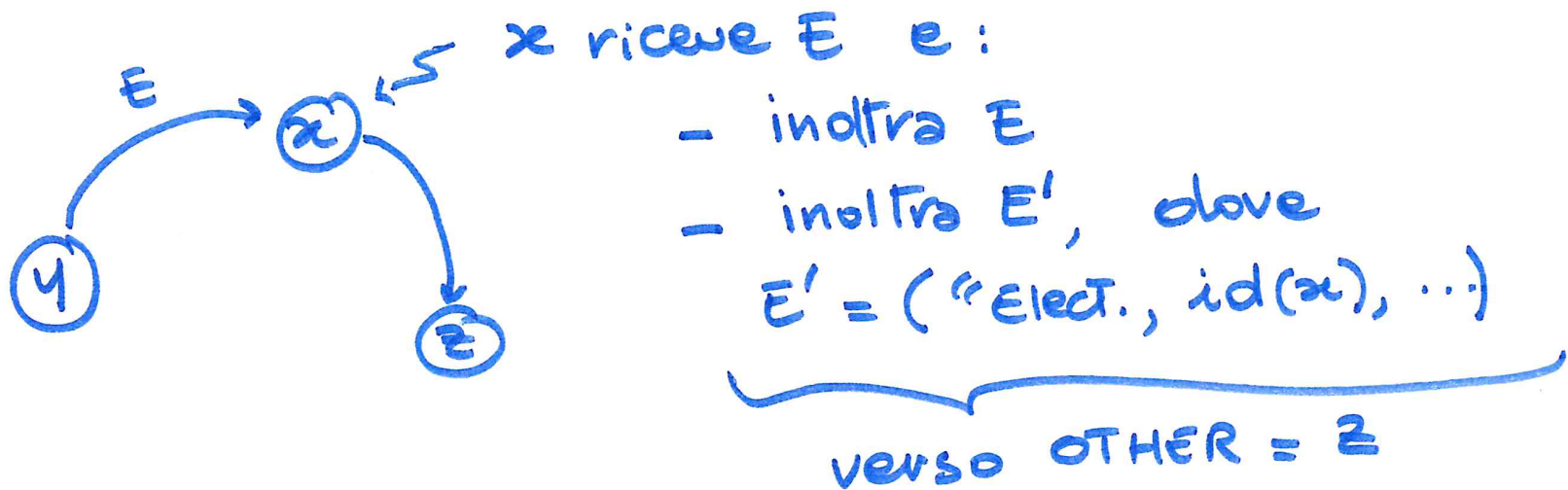
Notazione:  $N(x)$  - sender  
è detto OTHER



# Protocollo All The Way

I msg viaggiano intorno all'anello inoltrati dalle entità nella stessa direzione

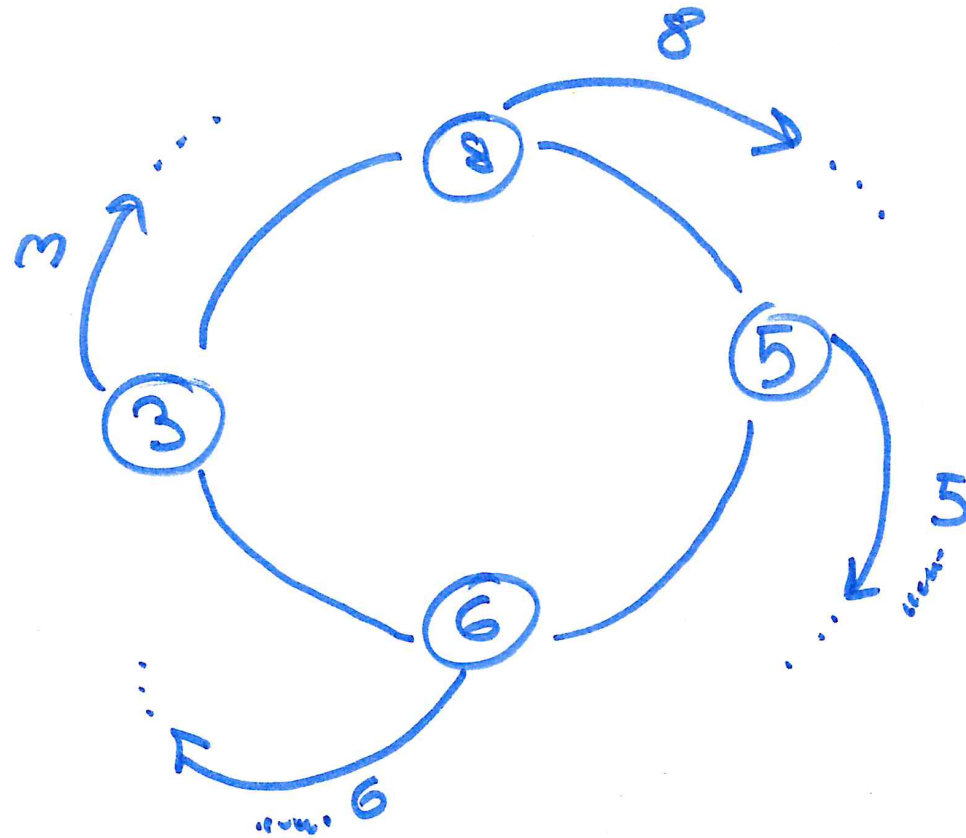
Tipo di msg: ("Elect.",  $id(x)$ , ...)



OSSERVIAMO:

Ogni entità  $x \in E$  vede  $id(y)$ ,  $\forall y \neq x \in E$ ,  
e può calcolare il minimo

Idea del protocollo: All the way



sull'anello  
viaggi 2  
ogni id!

Domanda: quando far Terminare una entità?

Risposta parziale:

Una volta che  $x$  riceve un msg  $E$  con il proprio  $id(x)$ , sa che  $E$  ha fatto il giro dell'anello  $\Rightarrow$  non lo inoltra più!

Può  $x$  Terminare?


SI

Solo se  
ne ha visti  $n$   
oliversi

NO

Si: Se si suppone la restrizione  
"Message Ordering" = prelievo dei msg  
Sui link secondo la politica FIFO  
ma noi non l'abbiamo!

Solo se  
ne ha viste  $n$ : Se si suppone che le entità siano  
a conoscenza della dimensione di  $A$   
ma noi non abbiamo questa restrizione!

No: Risposta corretta! Dobbiamo riempire  
in maniera opportuna i msg  $E$   
( $\text{Elect}, \text{id}(x), \dots$ )  qui per far Terminare  
correttamente le entità.



$E = (\text{"Elect."}, \text{id}(x), \text{counter})$

come usare counter?

- Inizialmente l'entità  $x$  pone  $\text{counter} = 1$
- Ogni altra entità  $y \neq x$  che inoltra  $E$  somma 1 a counter
- Quando  $E$  ritorna all'entità  $x$  counter è uguale a  $n = |A|$ ,
- Se  $x$  ~~riceve~~ ha già ricevuto  $n$  diversi id

$\Rightarrow$  può terminare

altrimenti aspetta  $\swarrow$  riceve altri msg e li inoltra, fa anche un controllo per verificare se è arrivato a  $n$  id  $\neq$

Sì

No

"All The Way"

STATI: { Asleep, awake, leader, follower }

STATINIT: { Asleep }

STATFINAL: { leader, follower }

Asleep

```
Spontaneously  
{ INITIALIZE;  
  become awake;  
}
```

Receiving ("Elect", value, counter)

```
{ INITIALIZE;  
  send ("Elect", value, counter+1)  
    To OTHER
```

```
  min = Min { min, value }  
  count = count + 1;  
  become awake;  
}
```

## Procedure INITIALIZE

```
{  count = 0;  
    size = 1;  
    know = false;  
    send ("Elect", id(x), size)  
        To right;  
    min = id(x);  
}
```

## Awake

Receiving ("Elect.", value, counter)

```
{ if value  $\neq$  id(x) then  
    { send ("Elect.", value, counter+1)  
        To other  
    }  
    min = Min {min, value};  
    count = count + 1;  
    if know = True Then CHECK;  
}  
else ...
```

```
else
{
    size = counter;
    know = true;
    CHECK;
}
}
```

Procedure CHECK;

```
{ if count = size then
    { if min = id(x) then
        become leader;
      else
        become follower;
    }
}
```



## Complessità di All the WAY

- Numero di msg:

$$M[\text{All the way} / \text{IR o Ring}] = n^2$$

Troppo costoso! Passiamo alla strategia b)

### Elect Minimum Initiator

" Solo gli initiator generano  $E$ , mentre le altre entità inoltrano i msg. "

### Problema di Terminazione:

da parte delle entità non initiator.

" quando gli Initiator hanno finito nel calcolo del leader mandano un msg di fine agli altri "  $= + n$  msg di fine

Complessità:

$$- M [\text{Alltheway} - \text{MinInitiator}] = n \cdot k + n$$

dove  $k = \text{n}^\circ \text{ di Initiator}$

$$- T [\text{Alltheway} - \text{MinInitiator}] \leq 3n - 1$$

spunto: si può raggiungere con 2 Initiator  
che si svegliano in momenti diversi!