

## Protocollo *Las Vegas*

Per ottenere il consenso mediante un protocollo *Las Vegas* è necessario aumentare la frazione di processi affidabili. Poniamo

$$n = 5t + 1, \quad L = \frac{n+1}{2} = 3t + 1 \quad \text{e} \quad H = L + t = 4t + 1.$$

### Algoritmo 4.2. *LVByzantineGeneral*

*Input:*  $b(i) = v_0(i)$

*Output:*  $b(i) = v$ .

---

*loop* = TRUE

while (*loop*)

1. trasmetti  $b(i)$  agli altri  $n - 1$  processi
  2. ricevi i valori spediti dagli altri  $n - 1$  processi
  3.  $maj(i) \leftarrow$  valore maggioritario tra i ricevuti (incluso il proprio)
  4.  $tally(i) \leftarrow$  numero dei valori uguali a  $maj(i)$
  5. if *testa*  
    then *soglia*  $\leftarrow L$   
    else *soglia*  $\leftarrow H$
  6. if  $tally(i) \geq \text{soglia}$   
    then  $b(i) \leftarrow maj(i)$   
    else  $b(i) \leftarrow 0$
  7. if  $tally(i) \geq 5t + 1$   
    then  $b(i) \leftarrow maj(i)$
- 

Analizziamo la correttezza di *LVByzantineGeneral* nei due casi possibili.

**Tutti i  $5t + 1$  processi affidabili sono inizializzati nello stesso modo:** supponiamo  $v_0 = 1$  per tutti i processi affidabili (per  $v_0 = 0$  vale lo stesso ragionamento). Al termine del primo *round* il consenso è stato raggiunto poiché se  $i$  è un qualunque processo affidabile, indipendentemente dall'operato dei  $t$  processi inaffidabili, avremo

$$maj(i) = 1, \quad tally(i) \geq 5t + 1 \quad \text{e} \quad b(i) = maj(i) = 1.$$

**Non tutti i processi affidabili sono inizializzati nello stesso modo:** abbiamo due sottocasi.

**Per ogni processo affidabile  $i$ ,  $maj(i)$  assume lo stesso valore:** dimostriamo che la malizia dei  $t$  processi inaffidabili può intervenire spedendo *bit* in modo che per alcuni processi il numero di valori uguali maggioritari sia inferiore e per altri superiore a  $L$  (o ad  $H$ ) ma non a tutte e due. Poiché  $H - L = t$ , se il numero di valori uguali maggioritari per alcuni è superiore ad  $H$ , allora è sempre superiore a  $L$ . Analogamente, se lo stesso numero per alcuni è inferiore a  $L$ , allora è sempre inferiore ad  $H$ . Pertanto, la scelta casuale della soglia vanifica l'azione dei processi *inaffidabili* il 50% delle volte. Osserviamo che nel primo caso la scelta della soglia  $L$  garantisce che il consenso sia uguale al valore maggioritario. Nel secondo, invece, il consenso è quello predeterminato (0 nella nostra implementazione).

**Per i processi affidabili  $i$  e  $j$ ,  $maj(i) \neq maj(j)$ :** mostriamo prima per assurdo che  $tally(i)$  e  $tally(j)$  devono entrambi essere minori di  $L$ . Aggiungiamo l'ipotesi  $tally(i) \geq L$ . Poiché ognuno dei processi affidabili spedisce lo stesso valore a tutti gli altri processi e al più tutti i  $t$  processi inaffidabili potrebbero aver spedito *bit* diversi a  $i$  e  $j$ , avremo

$$tally(j) \geq L - t = \frac{n+1}{2}.$$

Ma questo è assurdo poiché  $\text{maj}(i) \neq \text{maj}(j)$ .

Dal fatto che  $\text{tally}(i)$  e  $\text{tally}(j)$  sono entrambi minori di  $L$  segue che, indipendentemente dall'esito del lancio della moneta al passo 6. del primo *round*,  $\text{tally}(i)$  e  $\text{tally}(j)$  sono entrambi minori di *soglia*, il cui valore è  $L$  o  $H$ . Al termine del primo *round*, quindi, il *bit* di ogni processo affidabile vale 0 e il consenso sarà raggiunto all'iterazione successiva.

**Osservazione 4.1.** *Costo computazionale*

L'algoritmo *LVByzantineGeneral* converge in un numero atteso di *round* costante.

**Compito 4.1.** *Caso minimale di MCByzantineGeneral*

Considera il caso di un sistema distribuito costituito da  $n = 4$  processi di cui il quarto è inaffidabile. I tre processi affidabili seguono fedelmente il protocollo *Monte Carlo* mentre il processo inaffidabile, a ogni *round*, spedisce al processo affidabile  $i$  (con  $i = 1, 2$  e  $3$ ) il *bit*  $1 - b(i)$ . Implementa i tre processi affidabili e calcola media e varianza del numero di *round* necessari per raggiungere l'accordo. Determina empiricamente il numero di *round* dopo il quale la probabilità che l'accordo è raggiunto è più grande del 99.9% e cerca di dare una spiegazione al risultato ottenuto.