Principles of Cryptocurrency Design

Appunti ed Esercizi

Francesco Pasquale

26 marzo 2026

Nella lezione precedente abbiamo dimostrato che nessun protocollo deterministico per il problema *Byzantine Agreement* che termina in tempo finito può soddisfare *validity* e *consistency*.

Qui invece vediamo un protocollo probabilistico, essenzialmente una riformulazione del protocollo in [1], che con probabilità 1 termina in tempo finito e soddisfa *validity* e *consistency*.

1 Un protocollo per Byzantine Agreement nel modello asincrono

Algorithm 1 Async BA protocol [1]: Ogni nodo $u \in [n]$ esegue le istruzioni seguenti

Riceve in INPUT un bit $b \in \{0,1\}$.
$t = 1$
Invia $m = (b, t)$ a tutti
`decided` = FALSE
while `not decided` **do**
    when Arriva un messaggio $\hat{m} = (\hat{b}, h) \in \{0,1\} \times \mathbb{N}$ da un nodo $v \in [n]$
        if Hai già ricevuto un messaggio $(-, h)$ dal nodo v **then**
            Ignora $\hat{m}$
        else
            Aggiungi $\hat{m}$ all'insieme M_h, se esiste, altrimenti crea un insieme $M_h = \{\hat{m}\}$
        if $|M_t| \geqslant n - f$ **then**
            $v_0 = |\{(\hat{b}, t) \in M_t \,:\, \hat{b} = 0\}|$
            $v_1 = |\{(\hat{b}, t) \in M_t \,:\, \hat{b} = 1\}|$
            `decided` $= (\max\{v_0, v_1\} \geqslant n/2 + 3f + 1)$
            if $v_0 \geqslant n/2 + f + 1$ **then**
                $y = 0$
            else if $v_1 \geqslant n/2 + f + 1$ **then**
                $y = 1$
            else
                Scegli $y \in \{0,1\}$ u.a.r.
            $t = t + 1$
            Invia $m = (y, t)$ a tutti
OUTPUT y

Vogliamo dimostrare il teorema seguente.

Teorema 1. Se il numero di nodi corrotti è $f < (n-2)/10$ allora il protocollo asincrono in Algorithm 1 termina con probabilità 1 e soddisfa *validity* e *consistency*.

La dimostrazione passa per una serie di semplici passaggi, che qui lasciamo per esercizio con qualche suggerimento.

Esercizio 1. Dimostrare che, se esistono un $t \in \mathbb{N}$ e un $b \in \{0,1\}$ tali che tutti i nodi onesti inviano (b,t), allora tutti i nodi onesti danno in output b.
(Hint: Quando la condizione alla linea 11 *risulta* TRUE *per un certo t, degli $n-f$ messaggi in M_t almeno $n-2f$ provengono da nodi onesti. Quindi se ci sono un $t \in \mathbb{N}$ e un $b \in \{0,1\}$ per cui tutti i nodi onesti inviano (b,t), un nodo onesto u riceve almeno $n-2f$ messaggi (b,t). Quanto vale $n-2f$ se $f \leqslant (n-2)/10$? Quindi cosa fa il nostro nodo onesto u?)*

Si osservi che l'esercizio precedente implica che il protocollo soddisfa *validity*. Per quanto riguarda *consistency*, c'è da fare qualche altra osservazione.

Dato un nodo $u \in [n]$ e un $t \in \mathbb{N}$, chiamiamo *fase* t del protocollo per il nodo u le istruzioni dalla linea 12 alla linea 22. Si osservi che le *fasi* sono asincrone: un nodo u può essere nella fase t mentre un altro nodo v è nella fase $t' \neq t$.

Dato un nodo $u \in [n]$ e una fase $t \in \mathbb{N}$, indichiamo con $y_u(t)$ e $\texttt{decided}_u(t)$ le variabili locali y e `decided` del nodo u nella fase t. Diciamo che un nodo u imposta la sua variabile locale y *in modo deterministico* se il valore di y viene impostato nella linea 16 o nella linea 18.

Esercizio 2. Sia $t \in \mathbb{N}$. Dimostrare che se un nodo onesto u imposta la sua variabile $y_u(t)$ in modo deterministico, allora per ogni altro nodo onesto v che imposta la sua variabile $y_v(t)$ in modo deterministico deve essere $y_v(t) = y_u(t)$.
(Hint: Se un nodo onesto u imposta $y_u(t) = 0$ in modo deterministico (linea 16) allora ha ricevuto almeno $n/2 + f + 1$ messaggi $(0,t)$. Quanti messaggi $(1,t)$ al massimo può aver ricevuto un altro nodo onesto v?

Si osservi che l'esercizio precedente implica che, data una fase $t \in \mathbb{N}$, se per due nodi onesti u e v si ha che $y_u(t) \neq y_v(t)$ allora almeno uno dei due nella fase t ha impostato la sua variabile locale y in modo aleatorio (linea 20).

Esercizio 3. Osservare che dall'esercizio precedente segue anche che se due nodi onesti u e v impostano $\texttt{decided}_u(t) = \texttt{decided}_v(t) = \text{TRUE}$ allora $y_u(t) = y_v(t)$.

Esercizio 4. Siano $t \in \mathbb{N}$ e $b \in \{0,1\}$. Se un nodo onesto u imposta $\texttt{decided}_u(t) = \text{TRUE}$ e $y_u(t) = b$, allora ogni nodo onesto v invia il messaggio $(b, t+1)$.
(Hint: Se un nodo onesto u imposta decided$(t) = \text{TRUE}$ e $y_u(t) = b$ vuol dire che ha ricevuto almeno $n/2 + 3f + 1$ messaggi (b,t). Sia v un altro nodo onesto, quanti messsaggi (b,t) deve aver ricevuto v?

Esercizio 5. Per ogni $t \in \mathbb{N}$ tale che `decided` $=$ FALSE per tutti i nodi onesti, esiste un $b \in \{0,1\}$ tale che la probabilità che tutti i nodi onesti inviino $(b, t+1)$ è almeno 2^{-n}.
(Hint: Dall'Esercizio 2 segue che se due nodo onesti impostano la loro variabile y in modo deterministico, allora la impostano allo stesso valore. Quindi qual è la probabilità che tutti i nodi onesti che impostano la loro variabile y in modo aleatorio (linea 20) scelgano lo stesso valore dei nodi onesti che l'hanno impostata in modo deterministico?)

Proof. of Theorem 1. La proprietà di *validity* segue dall'Esercizio 1. Gli Esercizi 3, 4 e 1 implicano che se il protocollo termina allora soddisfa la proprietà di *consistency*.

Sia ora τ la variabile aleatoria che indica il primo $t \in \mathbb{N}$ in cui un nodo onesto $u \in [n]$ imposta $\texttt{decided}_u(t) = \text{TRUE}$

$$\tau = \inf\{t \in \mathbb{N} : \texttt{decided}_u(t) = \text{TRUE per qualche nodo onesto } u\}$$

Dall'Esercizio 5 segue che, per ogni $t \in \mathbb{N}$

$$\mathbf{P}(\tau > t) \leqslant \left(1 - 2^{-n}\right)^t$$

e quindi $\mathbf{P}(\tau < +\infty) = 1$. Dall'Esercizio 4 perciò segue che il protocollo termina con probabilità 1. □

Riferimenti bibliografici

[1] Michael Ben-Or. Another advantage of free choice (extended abstract) completely asynchronous agreement protocols. In *Proceedings of the second annual ACM symposium on Principles of distributed computing*, pages 27–30, 1983. https://dl.acm.org/doi/pdf/10.1145/800221.806707.