Principles of Cryptocurrency Design

Appunti ed Esercizi

Francesco Pasquale

28 marzo 2024

1 Byzantine Agreement

Nel problema *Byzantine Agreement (BA)* abbiamo n nodi di cui f sono corrotti. A ogni nodo i viene affidato in input un messaggio b_i. Vogliamo progettare un protocollo, che termini in un tempo finito, in cui ogni nodo onesto i dia in output un valore y_i tale che

- **Consistency**: Se i e j sono due nodi onesti, allora $y_i = y_j$;
- **Validity**: Se tutti i nodi onesti hanno in input lo stesso bit b, allora $y_i = b$ per ogni nodo onesto i.

Esercizio 1. Modificando opportunamente il protocollo di Dolev-Strong per il problema Byzantine Broadcast, mostrare che nel modello sincrono con PKI esiste un protocollo per Byzantine Agreement che soddisfa consistency e validity qualunque sia il numero di nodi corrotti.

Esercizio 2. Modificando opportunamente la dimostrazione di impossibilità del problema Byzantine Broadcast con $n/3$ o più nodi, mostrare che nel modello sincrono senza PKI non esiste nessun protocollo per Byzantine Agreement che soddisfa consistency e validity in presenza di $n/3$ o più nodi corrotti.

2 Modello asincrono

Nel modello asincrono non abbiamo un *global clock* e non abbiamo un *bound* sul tempo che impiega un messaggio inviato da un nodo a raggiungere il nodo di destinazione.

Un *protocollo* nel modello asincrono è *event driven*, cioè specifica cosa fa un nodo nel momento in cui riceve un messaggio: il nodo può eseguire un numero arbitrario di operazioni locali, cambiare *stato* (lo *stato* di un nodo è costituito dai valori delle sue variabili locali, che possono dipendere dall'input e da tutti i messaggi ricevuti). Una configurazione C è una coppia $C = (S, M)$ dove S rappresenta lo stato di ogni nodo e M è l'insieme di messaggi non ancora consegnati, la *message pool*. Un messaggio m è una coppia $m = (p, x)$ dove p è il nodo destinatario del messaggio e x è il contenuto del messaggio.

Se Π è un protocollo nel modello asincrono, data una configurazione $C = (S, M)$ e un messaggio $m = (p, x)$, indichiamo con $m(C)$ la configurazione $C' = (S', M')$ che si ottiene in base al protocollo quando il messaggio m viene consegnato, ossia l'insieme S' è quello che si ottiene da S aggiornando lo stato del nodo p dopo l'elaborazione delle informazioni contenute in x e M' è la *message pool* che si ottiene da M togliendo il messaggio appena consegnato m e aggiungendo tutti i messaggi inviati da p.

Esercizio 3. Osservare che, dato un protocollo deterministico Π, una configurazione C e due messaggi $m_1 = (p_1, x_1)$ e $m_2 = (p_1, x_2)$, se $p_1 \neq p_2$ allora $m_2(m_1(C)) = m_1(m_2(C))$.

Uno *schedule* σ è una sequenza ordinata di messaggi $\sigma = \big((p_1, x_1), \ldots, (p_k, x_k)\big)$. Dato un protocollo Π, una configurazione C e uno schedule σ, indichiamo con $\sigma(C)$ la configurazione che si ottiene partendo da C in base al protocollo Π dopo la consegna di tutti i messaggi in σ (nell'ordine stabilito dalla sequenza).

Esercizio 4. Osservare che, dato un protocollo deterministico Π, una configurazione C e due schedule $\sigma_1 = \big((p_1, x_1), \ldots, (p_k, x_k)\big)$ e $\sigma_2 = \big((q_1, y_1), \ldots, (q_h, y_h)\big)$, se gli insiemi di nodi destinatari nei due schedule sono disgiunti, $\{p_1, \ldots, p_k\} \cap \{q_1, \ldots, q_h\} = \emptyset$, allora $\sigma_2(\sigma_1(C)) = \sigma_1(\sigma_2(C))$.

3 FLP impossibility

A lezione abbiamo visto una dimostrazione del teorema seguente

Teorema 3.1 (Fischer, Lynch, Paterson [1])**.** Nel modello asincrono nessun protocollo deterministico per Byzantine Agreement che termina in un tempo finito può soddisfare consistency e validity, se c'è almeno un nodo corrotto.

Esercizio 5. Ripercorrere tutti i passaggi della dimostrazione nei dettagli, annotando eventuali punti poco chiari. Li discuteremo nella prossima lezione.

Riferimenti bibliografici

[1] Michael J. Fischer, Nancy A. Lynch, and Michael S. Paterson. Impossibility of distributed consensus with one faulty process. *Journal of the ACM (JACM)*, 32(2):374–382, 1985.