Principles of Cryptocurrency Design

Appunti ed Esercizi

Francesco Pasquale

17 marzo 2025

1 Byzantine Broadcast e State Machine Replication

Si consideri un sistema distribuito con n nodi in cui ogni nodo può eseguire computazioni locali arbitrarie e inviare messaggi di lunghezza arbitraria a ogni altro nodo. Degli n nodi, f sono *corrotti* e $n - f$ nodi sono *onesti*. Assumiamo di essere in un sistema *sincrono* in cui c'è un *global clock* noto a tutti i nodi che scandisce il tempo in *round* discreti e ogni messaggio inviato in un round t arriva a destinazione prima dell'inizio del round $t + 1$.

Consideriamo il problema seguente, che chiamiamo *State Machine Replication*: ad ogni nodo $i \in [n]$, in ogni round $r \in \mathbb{N}$, può essere affidata una o più *transazioni* `tx` (stringhe binarie). Ogni nodo i mantiene un *log* che consiste in una concatenazione di transazioni, indichiamo con LOG_i^r il log del nodo i al round r. Vogliamo progettare un protocollo che faccia in modo che l'evoluzione dei log dei nodi nel tempo soddifi le due proprietà seguenti

- **Consistency**: Per ogni $i, j \in [n]$ e per ogni coppia di round $r, s \in \mathbb{N}$, se i e j sono nodi onesti allora $\mathrm{LOG}_i^r \preccurlyeq \mathrm{LOG}_j^s$ oppure $\mathrm{LOG}_j^s \preccurlyeq \mathrm{LOG}_i^r$, dove con la notazione $\mathrm{LOG} \preccurlyeq \mathrm{LOG}'$ intendiamo che LOG è un prefisso di LOG'.
- **Liveness**: Esiste un *confirmation time* $T_{\mathrm{conf}} \in \mathbb{N}$ tale che, se una transazione `tx` viene affidata a un nodo onesto in un round $r \in \mathbb{N}$, allora per ogni nodo onesto $i \in [n]$, $\texttt{tx} \in \mathrm{LOG}_i^{r+T_{\mathrm{conf}}}$.

Esercizio 1. Mostrare che se abbiamo un protocollo Π_{BB} che risolve *Byzantine Broadcast* in R round e tollera fino a f nodi corrotti, allora possiamo progettare un protocollo Π_{SMR} per *State Machine Replication* con confirmation time $T_{\mathrm{conf}} = \mathcal{O}(nR)$ che tollera fino a f nodi corrotti.

Esercizio 2. Provare a progettare da zero un protocollo per *State Machine Replication* che tolleri $f \geqslant 1$ nodi corrotti e analizzarne correttezza (ossia dimostrare che il protocollo soddisfa *consistency* e *liveness*) e *confirmation time*.

Esercizio 3 (*Reading Proposal*)**.** Studiare il protocollo descritto in [2] e una sua implementazione pratica, descritta qui [1].

Riferimenti bibliografici

[1] Ben Auslin, Ertem Nusret Tas, Sheryl Hsu, and Ankur Agarwal. Streamlet: Optimizations, implementation and benchmarks. `https://www.scs.stanford.edu/24sp-cs244b/projects/Streamlet_Optimizations_Implementation_and_Benchmarks.pdf`, 2024.

[2] Benjamin Y Chan and Elaine Shi. Streamlet: Textbook streamlined blockchains. In *Proceedings of the 2nd ACM Conference on Advances in Financial Technologies*, pages 1–11, 2020. `https://dl.acm.org/doi/pdf/10.1145/3419614.3423256`.