ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

SCUOLA DI SCIENZE

Corso di Laurea Triennale in Informatica

An analysis of Proof-of-X blockchain consensus protocols

Relatore:

Chiar.mo Prof.

Cosimo Laneve

Presentata da:

Erik Montalbetti

Correlatore**:**

Dott.ssa

Adele Veschetti

II Sessione
Anno accademico 2018-2019

Prefazione

Questo elaborato viene redatto con lo scopo di presentare un'analisi di alcuni dei protocolli per il consenso distribuito ad oggi esistenti nel panorama della tecnologia blockchain.
In particolare, viene presentata la strategia denominata *Proof-of-Work* e vengono esaminate alcune alternative di più recente comparsa: *Proof-of-Stake*, *Delegated Proof-of-Stake* e *Proof-of-Authority*.
Infine, si mostra un confronto dei suddetti protocolli dal punto di vista delle caratteristiche maggiormente desiderabili e delle prestazioni di un sistema blockchain.

Indice

Capitolo 1

Introduzione

La blockchain è una tecnologia introdotta nel 1991 da un'idea dei ricercatori Stuart Haber e W. Scott Stornetta [19], che ha acquisito importanza in seguito all'ideazione nel 2008 e all'implementazione nel 2009 del sistema *Bitcoin* per mano di Satoshi Nakamoto [26].
Questa tecnologia ha recentemente catalizzato una crescente attenzione ed ha assunto una posizione sempre più significativa in ambito accademico e imprenditoriale, tanto da essere paragonata ad Internet in quanto a capacità di rivoluzionare la vita umana.
Molti ne parlano idolatrandola, affascinati dalle innovazioni che porterà in una moltitudine di settori, altri ne sottolineano i rischi per la sicurezza arrivando a definirla addirittura una tirannia. Quello che è certo è che ad oggi ci troviamo in una fase a cavallo tra studio e sperimentazione della blockchain e che questa sta guadagnando un sempre maggiore consenso, soprattutto nel mondo delle imprese impegnate nel processo di innovazione, per la sua capacità di ridurre i costi e migliorare l'efficienza.
Il concetto di blockchain viene spesso legato al mondo della finanza e alle opportunità di indipendenza economica ottenibile mediante l'adozione delle criptovalute, di cui i Bitcoin sono l'espressione più conosciuta. Tuttavia, questa recente tecnologia trova interessanti applicazioni anche in altri campi, come ad esempio [27]:

- *Internet of Things* (*IoT*): blockchain può veicolare le informazioni raccolte da dispositivi connessi e rendere più veloci e più sicuri i processi in cui questi sono coinvolti;
- *Industria 4.0*: grazie alle sue caratteristiche, blockchain può semplificare i processi aziendali e incrementare l'efficienza dell'intero sistema;
- *Servizi di sicurezza*: privacy, controllo degli accessi, autenticazione per applicazioni distribuite, gestione sicura dei dati;
- *Agrifood*: molte aziende utilizzano già blockchain per la tracciabilità dei prodotti agroalimentari lungo tutta la filiera e, ad esempio, un recente progetto promosso dal MISE si propone di sfruttare questa tecnologia per la tutela del *Made in Italy*;

- *Servizi pubblici*: blockchain semplifica la gestione dei processi amministrativi della pubblica amministrazione;
- *Smart contracts*: distribuiti su una blockchain implementano la stipula di un contratto digitale tra parti che non si conoscono e non possono fidarsi l'una dell'altra, consentendo l'esecuzione dei termini contrattuali al verificarsi di determinati eventi [36];
- *Sanità*: diversi sistemi sanitari nel mondo adottano blockchain per la registrazione dei dati dei pazienti e delle loro cartelle cliniche [4];

e più in generale in ambiti in cui vi sia necessità di immutabilità dei dati, trasparenza e/o fiducia tra i partecipanti.

1.1 Struttura della tesi

La tesi è strutturata nel seguente modo: dopo aver presentato, nel primo capitolo, una panoramica sulla tecnologia blockchain, nel secondo capitolo se ne fornisce una definizione più tecnica. All'interno dello stesso capitolo vengono illustrati gli elementi di crittografia alla base di questa tecnologia e si entra nel dettaglio del suo funzionamento, analizzandone le principali caratteristiche.
Nel capitolo 3 vengono definiti i protocolli di consenso e si analizza il più utilizzato, *Proof-of-Work* con alcune considerazioni relative alla sua sicurezza.
In modo analogo, si procede allo studio dei protocolli *Proof-of-Stake*, *Delegated Proof-of-Stake* e *Proof-of-Authority*, rispettivamente nei capitoli 4, 5 e 6.
Infine, nel capitolo 7 vengono messe a confronto le quattro strategie di consenso esposte.

Capitolo 2

Il paradigma blockchain

Nel mondo delle transazioni di beni e servizi il paradigma tradizionale prevede la presenza di una autorità centrale, a cui gli utenti si affidano perché faccia da intermediaria negli scambi con altri partecipanti al medesimo sistema, con i quali non esiste un rapporto di fiducia. Ad esempio, quando si effettua un acquisto con carta di credito, entrambe le parti in gioco si affidano ad una banca, affinché questa si occupi di verificare l'effettiva disponibilità economica dell'acquirente e di portare a compimento lo scambio di denaro. La presenza di una autorità centrale, nel precedente esempio rappresentata dalla banca, garantisce la corretta esecuzione delle transazioni, ma introduce due problemi: l'accentramento del potere nella autorità stessa e la presenza di un *single point of failure* nel sistema.

La blockchain si propone come alternativa al paradigma centralizzato, risolvendo questi problemi. Essa può essere vista come un registro condiviso e distribuito, un database cui l'accesso e le modifiche sono consentiti ai partecipanti distribuiti su una rete peer-to-peer.

In quanto registro distribuito, la blockchain non esiste in un'unica istanza centralizzata, ma piuttosto in più "copie" sincronizzate in possesso dei partecipanti alla rete. Questi ultimi sono detti nodi e sono fisicamente costituiti dai server attraverso cui sono gestite le transazioni. In questo modo i poteri e i diritti sul registro sono equamente suddivisi tra tutti i nodi e viene eliminato il problema del *single point of failure*, poiché anche nel caso in cui qualche nodo dovesse guastarsi, non verrebbe compromesso il funzionamento dell'intero sistema.

2.1 Crittografia

Per garantire la sicurezza e l'integrità dei dati in un sistema distribuito, la tecnologia blockchain sfrutta tecniche di crittografia già consolidate. In particolare, essi fanno uso di funzioni di *hash* e di crittografia asimmetrica.

Gli algoritmi di *hash* sono da tempo impiegati in molte applicazioni di carattere informatico, soprattutto per la creazione e la verifica di firme digitali e l'analisi dell'integrità dei dati.
Una funzione di *hash H* è tale che [24]:

- *H* è applicabile a diversi argomenti di diversa dimensione;
- Dati *H* e un input *x*, *H*(*x)* produce sempre lo stesso risultato, detto *digest* o *hash* di x;
- Dati *H* e un input *x*, *H*(*x*) ha dimensione fissa
- Dati *H* e un input *x*, è semplice calcolare *H*(*x*)
- È computazionalmente difficile invertire la funzione *H*, ovvero trovare un input *x* partendo dal suo *digest*.

Uno degli algoritmi di *hash* più utilizzati, anche in ambito blockchain, è *SHA-256* (*Secure Hash Algorithm-256*) che produce un *digest* di 256 bit.
La tecnica di crittografia asimmetrica è utilizzata nella comunicazione tra mittente e destinatario di un messaggio. Ad ogni partecipante alla comunicazione sono associate una chiave privata, nota unicamente al proprietario, e una chiave pubblica, che è appunto di pubblico dominio.
I dati cifrati con una chiave pubblica possono essere decifrati solo con la relativa chiave privata e viceversa. Questa configurazione apre la strada a due applicazioni:

- La cifratura di dati che necessitano di segretezza: chiunque con la chiave pubblica di un destinatario può cifrare un messaggio che solo il destinatario stesso, con la sua chiave privata, sarà in grado di decifrare;
- La cosiddetta firma digitale: l'identità del mittente di un messaggio è verificabile usando la sua chiave pubblica per la decifrazione. Se questa ha successo significa che il messaggio proviene effettivamente da chi ci si aspetta. Infatti, solo il possessore della chiave privata, associata alla chiave pubblica utilizzata, può aver effettuato la cifratura.

2.2 L'interno di una blockchain

L'entità atomica che costituisce la blockchain è rappresentata dalla singola transazione, che altro non è se non una struttura dati che veicola le informazioni relative allo scambio espresso dalla transazione stessa.
Le transazioni sono a loro volta raggruppate ed inserite all'interno di un'altra struttura dati chiamata blocco, definita dai seguenti campi:

- *Blocksize*: dimensione del blocco, può variare a seconda dell'implementazione della blockchain;
- *Blockheader*: formato da diversi campi tra cui il *timestamp* del blocco e *l'hash* del blocco precedente;
- *Transaction counter*: numero di transazioni contenute nel corpo del blocco;
- *Transactions*: il corpo del blocco, formato da una lista non vuota di transazioni.

Il primo blocco di una blockchain è detto *genesis block*. Una blockchain è dunque una catena di blocchi contenenti transazioni.
Per poter aggiungere una transazione alla blockchain, un nodo deve creare la transazione e propagarla agli altri nodi del network, in modo che questi possano verificarla. La verifica di una transazione è una constatazione della sua effettuabilità.
Si supponga, ad esempio, che Greta voglia trasferire una parte del suo *cripto-asset* a Linda, dove per cripto-asset si intende la rappresentazione digitale di una "unità di valore" su una generica blockchain (ad esempio una criptovaluta). Greta crea dapprima la transazione e vi applica la funzione di *hash* per calcolarne il *digest*. A questo punto, usando la sua chiave privata, Greta firma il *digest*, ottenendo la firma digitale della transazione. Infine, riporta la chiave pubblica di Linda e propaga nella rete la transazione così generata, affinché possa essere verificata dagli altri nodi.
La fase di verifica di una transazione è detta di "verifica indipendente", perché ogni nodo che riceve la transazione controlla la sua correttezza indipendentemente da tutti gli altri, consultando la propria copia della blockchain.

Un gruppo di transazioni verificate viene inserito in un blocco da un nodo eletto in base al protocollo della blockchain. L'attività di creazione di un blocco è detta validazione. Un blocco validato viene inviato al network e viene finalizzato, cioè considerato irrevocabilmente parte della catena, solo quando raggiunge una distanza *k* dall'ultimo nodo, ovvero dopo l'aggiunta di altri *k* blocchi (in Bitcoin *k* = 6) [26].

Quando più blocchi vengono validati nello stesso momento, può capitare che si assista ad una ramificazione della blockchain. Questo accade perché alcuni dei partecipanti alla rete possono scegliere di lavorare su uno dei blocchi appena aggiunti, mentre altri potrebbero sceglierne un altro, continuando la catena in una direzione diversa. Tale situazione, nota come "*fork* accidentale", è dovuta all'inevitabile ritardo di propagazione del blocco ai nodi della rete.

Lo scenario appena descritto può profilarsi anche nel caso di un attacco al sistema: un nodo malintenzionato può deliberatamente causare una *fork*, nel tentativo di modificare transazioni precedentemente inserite nella catena.

La risoluzione di una *fork* è graduale e consiste nella scelta di uno solo dei rami creatisi, con l'abbandono di quelli alternativi, che diventano "orfani".

Tutte le operazioni finora descritte concorrono a formare il cosiddetto meccanismo di consenso, per cui tutti i nodi convengono nel giudicare validi i blocchi da concatenare. Le modalità di svolgimento di dette operazioni sono disciplinate dalle regole che costituiscono il protocollo di consenso della blockchain.

In base alla gestione prevista in ciascun sistema, si possono distinguere:

- Blockchain pubbliche o *permissionless*: a cui ognuno può prendere parte come nodo, i record sono visibili a tutti e ognuno partecipa al processo per il raggiungimento del consenso;
- Blockchain private o *permissioned*: dove il consenso è raggiunto con la partecipazione dei nodi dell'organizzazione che gestisce la blockchain e vi sono restrizioni sui diritti di scrittura e lettura della catena.

2.3 Caratteristiche principali

Si è visto come funziona al suo interno una blockchain, si possono dunque riassumere le sue principali caratteristiche in:

- *Decentralizzazione*: blockchain non risponde al paradigma dei database centralizzati, la cui esistenza è fisicamente legata ad un server con cui è possibile interagire mediante determinate richieste regolate da protocolli client-server e in cui le transazioni sono validabili da un'unica entità centrale. Al contrario blockchain è accessibile ad ogni nodo in lettura, dal momento che è presente una copia del database presso ciascun partecipante alla rete, ma anche in scrittura, secondo regole dettate dai protocolli di consenso.
- *Immutabilità e sicurezza*: per via della sua stessa natura è impossibile modificare o eliminare dati memorizzati in blocchi già validati e concatenati, poiché si rivelerebbe necessario violare ogni singola copia della blockchain in possesso dei nodi partecipanti. Inoltre, mediante tradizionali tecniche crittografiche e di firma digitale, vengono verificate le identità dei protagonisti di una transazione e il contenuto della stessa.
- *Tracciabilità e trasparenza*: chiunque, in qualsiasi momento è in grado di risalire alla creazione e alla storia di ogni transazione registrata indelebilmente nei blocchi della catena.

Capitolo 3
Protocolli di consenso distribuito

In una blockchain il problema principale consiste nella definizione del successivo blocco da concatenare, operazione che, come accennato, prende il nome di validazione. Questo compito e la relativa ricompensa vengono assegnati ad un nodo eletto in base al protocollo di consenso distribuito.

Il protocollo garantisce la sicurezza del sistema e la coerenza dei dati presenti nel registro. Esso si declina in algoritmi diversi a seconda della blockchain considerata.

Questi algoritmi si servono di tecniche crittografiche per regolamentare la gestione delle operazioni di concatenamento e la risoluzione delle situazioni di conflitto (che portano ad una ramificazione della catena), salvaguardando il sistema da eventuali attaccanti. Va infatti sottolineato come la stessa presenza di una ricompensa prevista per i validatori sia un incentivo per tutti i partecipanti a mantenere un comportamento corretto.

Il protocollo di consenso distribuito è dunque il cuore dell'intero sistema, dal momento che impatta direttamente sull'efficienza dello stesso.

3.1 Proof-of-Work

Il protocollo ad oggi più utilizzato nelle blockchain prende il nome di *Proof-of-Work* (*PoW*). L'idea alla base di questa strategia è stata proposta per la prima volta nel 1992 da Cynthia Dwork e Moni Naor [13] come metodo per porre un freno al fenomeno delle e-mail indesiderate, anche se il concetto è stato formalizzato solo nel 1999 da Markus Jacobsson e Ari Juels, che lo applicarono al contesto dei protocolli crittografici [20].

3.1.1 Funzionamento

In PoW i nodi partecipanti competono tra loro per diventare *leader*, ovvero per guadagnare il diritto di aggiungere il prossimo blocco alla catena, cercando di risolvere un problema computazionale, che consiste in un puzzle crittografico.
Il nodo che, utilizzando le risorse a sua disposizione, presenta per primo la soluzione, viene ricompensato per il lavoro svolto. Questo lavoro viene denominato *mining* e i nodi validatori sono pertanto *miners.* Per favorire una velocità accettabile nel processo di validazione, mantenendo allo stesso tempo un buon grado di sicurezza, la soluzione al problema è pensata in modo da essere computazionalmente complessa da trovare, ma di facile verifica.
Per diventare leader, ogni nodo è costantemente impegnato nella computazione di un valore generato come output di una funzione *hash*, che prende in input l'*header* del nuovo blocco, costituito dai seguenti campi:

- *Hash* del blocco precedente;
- Numero di versione del blocco;
- *Timestamp* del blocco;
- *Merkle root* delle transazioni: una breve rappresentazione delle transazioni contenute nel blocco proposto;
- *Nonce*: un numero esadecimale casuale, da modificare ad ogni tentativo.

Un nodo guadagna la possibilità di validare il blocco proposto quando il valore di *hash* da esso computato è minore o uguale ad una soglia detta *target*, ovvero:

$$Hash(Block) \leq target \tag{3.1}$$

dove *Block* rappresenta, per convenzione, la concatenazione degli argomenti di input della funzione *hash* sopra elencati.
La ricerca di una soluzione al puzzle crittografico si riduce quindi alla ricerca del *nonce* per cui sia verificata la (3.1). Ciò è possibile iterando su valori diversi del *nonce* fino ad arrivare ad un valore di *hash* che rientra nel range richiesto.

Il target ha un valore fissato nel *genesis block* e viene modificato ogni 2016 blocchi validati da un algoritmo che agisce come segue [15]:

- Se gli ultimi 2016 blocchi sono stati validati mediamente in meno di 10 minuti, si riduce il valore del target;
- se gli ultimi 2016 blocchi sono stati validati mediamente in più di 10 minuti, si aumenta il valore del target.

Ovviamente, minore è il valore del target, minore sarà il range di valori di *hash* validi e di conseguenza maggiore sarà la difficoltà nel generare un nuovo blocco.
In una rete di *N* nodi, la probabilità p_i del nodo *i* di essere eletto per minare il nuovo blocco è [27]:

$$p_i = \frac{c_i}{\sum_{j=1}^{N} c_j} \tag{3.2}$$

dove c_i è l'*hashrate* del nodo *i*, ovvero la misura della sua potenza computazionale.

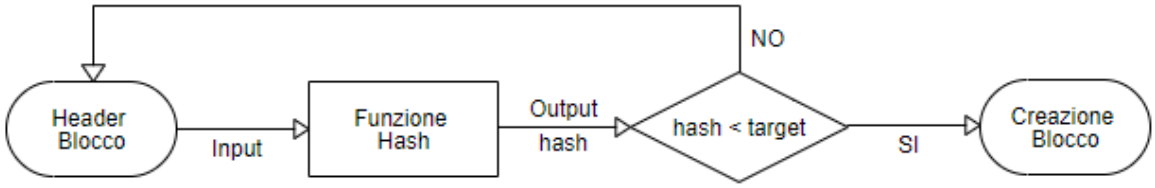

Figura 3.1: Flusso PoW.

3.1.2 Considerazioni

Il grado di difficoltà dell'enigma da risolvere in PoW per ricevere la ricompensa derivante dal mining deve innalzarsi al crescere della capacità di lavoro della rete. Man mano che la rete si espande, questo si traduce in un considerevole e crescente dispendio di risorse, dal momento che ogni partecipante tende a voler incrementare il proprio *hashrate* per avere maggiore probabilità di diventare leader e per farlo sono richiesti hardware specializzato ed energia elettrica per alimentarlo [17]. Se da un lato questo comporta una scarsa efficienza del sistema,

dall'altro rende onerosi gli attacchi al sistema, fungendo da misura di sicurezza. Infatti, per eseguire un attacco, ad esempio su una blockchain come quella di Bitcoin, un nodo (o un gruppo di nodi) malintenzionato dovrebbe riuscire ad ottenere più della metà della potenza di calcolo di tutti i validatori. In questo modo otterrebbe la possibilità di imporre come veritiera la propria versione della catena di blocchi, volgendo a proprio vantaggio la politica di gestione prevista dal protocollo nel tentativo di alterare le transazioni. Questa prospettiva, conosciuta come *51% attack* o *majority attack*, faciliterebbe il cosiddetto *double spending*: un nodo malintenzionato potrebbe, ad esempio, registrare una transazione e poi propagare una versione della blockchain in cui quella transazione non compare, potendo così spendere più volte la stessa moneta.

Grazie alla grandezza della sua rete, Bitcoin non ha mai subìto un *51% attack*, che è invece riuscito contro alcune *altcoin* (criptovalute alternative) [25].

Nel tempo sono state proposte diverse alternative a PoW, tra cui ad esempio Proof-of-Capacity, Proof-of-Burn, Proof-of-Elapsed-Time, Proof-of-Stake, Delegated Proof-of-Stake e Proof-of-Authority. Nel seguito si prenderanno in considerazione gli ultimi tre modelli annoverati.

Capitolo 4
Proof-of-Stake

Uno dei principali candidati a rimpiazzare PoW è il protocollo *Proof-of-Stake* (*PoS*).
La strategia alla base di PoS è stata proposta per la prima volta nel luglio del 2011 da un utente del forum *Bitcointalk* [31], proprio come alternativa alla PoW, ed è stata formalizzata nel 2012 in un elaborato a cura di Sunny King e Scott Nadal [22], fondatori della blockchain di *Peercoin*, anche nota come *PPCoin*. Recentemente anche *Ethereum*, una delle blockchain più estese e conosciute, ha avviato la procedura di transizione dal protocollo PoW a PoS.

4.1 Funzionamento

Nei sistemi basati su Proof-of-Stake il consenso viene raggiunto tramite selezione pseudo-casuale del leader incaricato della creazione del successivo blocco. Mentre questa scelta in PoW si basa unicamente sulla potenza di calcolo dei nodi validatori, in PoS essa avviene in funzione dell'interesse (*stake*) di ciascun nodo nel prendere parte al sistema. Lo *stake* in PoS è generalmente rappresentato dall'ammontare di criptovaluta impegnato come garanzia da ogni partecipante, anche detto *stakeholder*. Maggiore è questo valore, maggiore è l'interesse del nodo a mantenere un comportamento corretto e una migliore solidità della rete e maggiore è la probabilità che il nodo in questione venga eletto come leader. Si può quindi dire che la probabilità p_i che il nodo i venga eletto leader in una rete con N partecipanti è:

$$p_i = \frac{s_i}{\sum_{j=1}^{N} s_j} \tag{4.1}$$

dove s_i è lo *stake* del nodo i.
Come previsto da PoW, anche in PoS il validatore ha diritto al riconoscimento di una ricompensa, in questo caso sotto forma di onorario.

In PoS l'attività di creazione di un nuovo blocco è spesso detta *minting* (coniazione) o *forging* (forgiatura) e un validatore è un *minter* o *forger*.
Per evitare che venga sempre selezionato il nodo più "ricco" della rete e garantire la casualità della scelta, in aggiunta allo *stake* di ogni nodo, contribuiscono all'elezione ulteriori criteri, che cambiano da un'implementazione all'altra di PoS.
I criteri più spesso utilizzati sono due:

- *Randomized Block Selection*: il leader è scelto sulla base di una formula che combina il totale dello *stake* di un nodo con informazioni relative all'ultimo blocco della catena.
  In questo tipo di implementazione di PoS, ad ogni blocco è associato un campo denominato *generation signature*. Ogni nodo firma questo parametro con la propria chiave pubblica e ne calcola *l'hash*. Vengono considerati i primi 8 byte del valore ottenuto, denominati *hit* del nodo. Il nodo stesso calcola un valore *target* nel seguente modo [28]:

$$target = S * B_e * target_b \tag{4.2}$$

  dove *S* è il tempo trascorso dalla validazione dell'ultimo blocco, B_e è lo *stake* effettivo del nodo e $target_b$ è un valore chiamato *base target*, che è calcolato in funzione dei metadati dell'ultimo blocco e regola la velocità di validazione della rete.
  Affinché un nodo venga eletto come validatore, questo algoritmo prevede che:

$$hit < target \tag{4.3}$$

  Il valore del target cresce con il passare del tempo e con esso cresce il range dei valori di *hit* per i quali vale la (4.3). In questo modo, anche se ci sono pochi nodi attivi sulla rete, uno di essi prima o poi genererà un blocco.

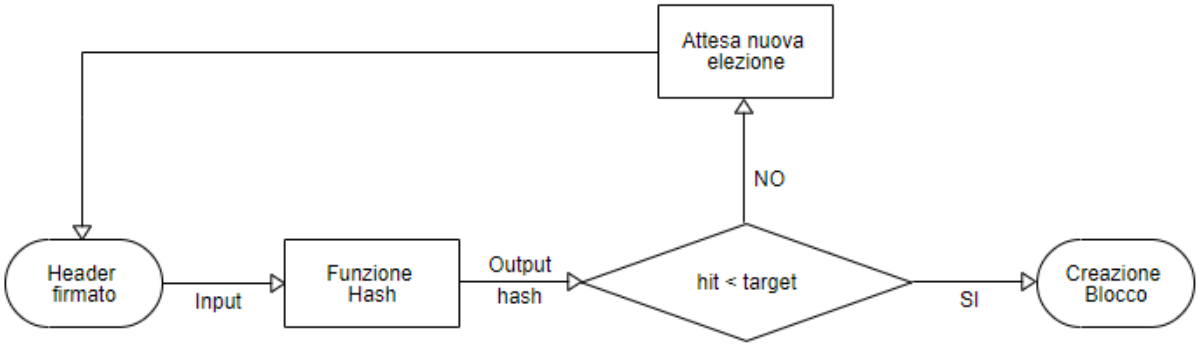

Figura 4.1: Flusso PoS con Randomized Block Selection.

- *Coin Age-based Selection*: si basa sul concetto di invecchiamento (utilizzato già nella blockchain di Bitcoin per assegnare priorità alle transazioni nella rete) e prevede che lo *stake* di ciascun nodo venga moltiplicato per il numero di giorni in cui è rimasto invariato, cioè non è stato speso (generalmente il minimo è 30 giorni). Il valore così ottenuto è detto *coin age* e rappresenta la probabilità del nodo di essere scelto come leader. Quando ciò accade la sua *coin age* viene azzerata.

  A titolo di esempio, se Greta ha uno *stake* di 10 monete inutilizzate da 20 giorni, il suo valore di *coin age* è 200. Se decidesse di inviare anche solo una parte delle sue 10 monete a Linda, la sua *coin age* verrebbe consumata e verrebbe aggiornata a 0. Lo stesso vale quando Greta viene eletta come leader.

  In base a questo criterio, ad ogni elezione viene scelto il nodo che calcola un valore che soddisfi la condizione:

$$Hash(Blocco) < coinage * target \quad (4.4)$$

  dove *Blocco* è la concatenazione degli argomenti *timestamp*, *hash* del blocco precedente, indirizzo del nodo candidato.

Esistono alcune implementazioni di PoS che adottano una combinazione dei due metodi appena visti.

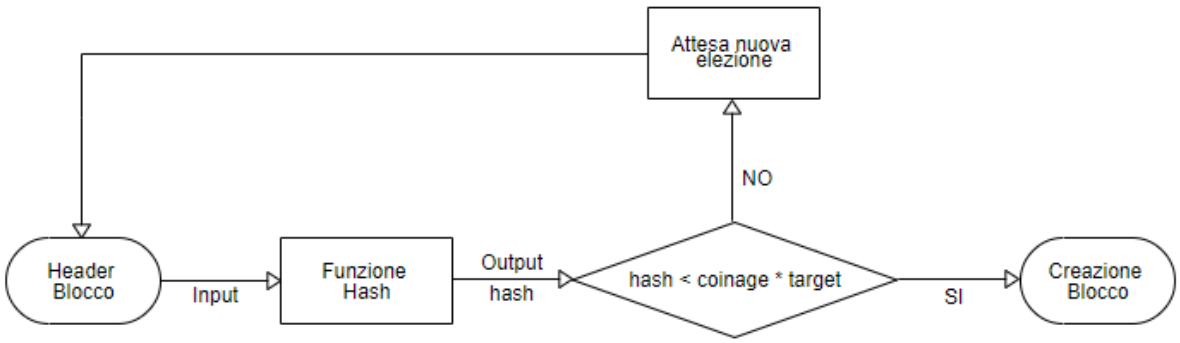

Figura 4.2: Flusso PoS con Coin Age-based Selection.

4.2 Considerazioni

Si è già accennato a come il corretto funzionamento di PoW e la sua sicurezza siano dipendenti da un elevato consumo di energia, destinato a crescere esponenzialmente. Nel tempo questo ha portato numerosi *miners* a riunirsi in gruppi che condividono le risorse, con l'obiettivo sia di massimizzare i guadagni, sia di risparmiare sui costi. Questi gruppi, detti *mining pools,* arrivano talvolta ad investire milioni di dollari nell'attività di *mining*. Conseguenze dirette di questo fenomeno sono una riduzione della decentralizzazione e un aumento del rischio che questi gruppi possano effettuare un *51% attack* ai danni della rete.

PoS si propone come alternativa più efficiente e più sicura. I sistemi basati su PoS, infatti, richiedono un quantitativo di energia paragonabile a qualsiasi altro protocollo Internet già esistente e non richiedono l'impiego di hardware specializzato. Inoltre, non essendovi alcun legame tra il protocollo e le risorse del mondo reale, il rischio della formazione di *mining pools* e della conseguente centralizzazione del sistema è ridotto.

PoS è meno esposto di PoW a *51% attacks*: per poter monopolizzare una blockchain che adotta questo tipo di protocollo è necessario entrare in possesso del 51% dello *stake* totale sulla rete, impresa che può essere molto costosa. A rendere svantaggioso un attacco di questo tipo vi è anche il fatto che se un nodo riuscisse ad entrare in possesso di più della metà del cripto-asset di una rete, i rimanenti partecipanti tenderebbero ad uscirne, facendo precipitare il valore dello *stake* acquisito dall'attaccante.

I detrattori di PoS ritengono che esistano delle vulnerabilità non trascurabili e nuove rispetto a PoW, che non lo rendono il protocollo ideale da adottare, in particolare la possibilità di attacchi *nothing at stake* e *long-range attack.*

Un attacco *nothing at stake* si verifica quando, in caso di *fork*, i nodi cercano di creare intenzionalmente nuovi blocchi su più di un ramo della catena, per aumentare i guadagni potenziali. Un possibile espediente per arginare questo tipo di minaccia è l'introduzione di un meccanismo di penalizzazione per i nodi scorretti, come proposto in *Snow White* [10]. In alternativa è possibile evitare il verificarsi di *fork*, come accade in *Algorand* [18].

In un *long-range attack* un attaccante corrompe altri *k* partecipanti per avere le loro chiavi private, utilizzate in passato in fase di creazione di *k* blocchi. Usando queste chiavi l'attaccante può provare a riforgiare quei blocchi, riscrivendo a suo piacimento la storia della blockchain.

Anche per questa modalità di attacco sono state proposte diverse tecniche di mitigazione.

Nonostante esistano diverse implementazioni basate su PoS che si propongono di risolvere le criticità relative alla sicurezza, il loro sviluppo è ancora in fase embrionale ed è necessario approfondire e testare la solidità delle diverse soluzioni [3, 16].

Capitolo 5

Delegated Proof-of-Stake

Il protocollo *Delegated Proof-of-Stake* (*DPoS*) è un'estensione di PoS, proposta nel 2014 da Daniel Larimer, sviluppatore e fondatore delle piattaforme blockchain di *BitShares*, *EOS* e *Steem*. La sua prima implementazione ha visto la luce nel 2015 e oggi DPoS è utilizzata anche da diverse altre piattaforme blockchain, quali *Tron*, *Cardano* e *Tezos*.

5.1 Funzionamento

Nella strategia di DPoS l'intero processo per il raggiungimento del consenso è suddiviso in due parti: la prima consiste nell'elezione di un gruppo di nodi creatori ad opera dell'intera comunità della rete, la seconda nell'effettiva creazione dei blocchi. Chiunque può decidere di partecipare al processo di validazione, candidandosi e convincendo la comunità di partecipanti di avere i requisiti per svolgere il proprio compito senza causare intoppi al sistema.
I candidati che vengono eletti sono detti "*testimoni*" (*witnesses*) e comprendono un sottoinsieme della totalità dei partecipanti, generalmente tra 21 e 101 nodi. In modo simile a quanto avviene in PoS per la scelta del leader, in DPoS è lo *stake* dei partecipanti a determinare il peso del loro voto. A titolo di esempio, se Greta e Linda hanno uno *stake* rispettivamente di 20 e 10 monete, nel caso in cui la prima votasse per il nodo A e la seconda votasse per il nodo B, sarebbe A ad avere la meglio. In pratica è come se venisse assegnato un punteggio ai candidati in base allo *stake* dei votanti: in questo esempio A riceve 20 punti, B ne riceve 10.
Al termine di ogni votazione, che avviene ad intervalli prestabiliti (su BitShares ogni 24 ore), vengono selezionati come testimoni i primi *N* nodi che hanno raggiunto il punteggio più alto sulla base del meccanismo appena mostrato, dove *N* è un parametro specifico del sistema.
I testimoni eletti vengono disposti in ordine casuale e viene loro richiesto di produrre e proporre un blocco in un tempo prefissato. Quando viene il suo turno ogni testimone *i* crea un blocco con le transazioni ricevute dalla rete, lo firma e lo propaga a tutti i partecipanti.

Se un delegato non riesce a produrre un blocco nel suo turno (ad esempio perché è andato offline o non completa in tempo la procedura), l'evento viene registrato dal sistema e può portare, previa votazione della rete, all' espulsione del nodo dal gruppo dei testimoni. Quando l'*N*-esimo di questi crea il suo blocco, l'ordine viene rimescolato e il ciclo ricomincia.

La creazione di ogni blocco è ricompensata sulla base di un parametro del sistema, in aggiunta ad un eventuale onorario relativo ad ogni transazione.

Un aspetto importante di DPoS è dato dal fatto che, sfruttando il processo di votazione, i parametri della rete possono essere modificati in qualunque momento senza causare una *fork*. Oltre ai testimoni esiste infatti un altro gruppo di nodi delegati, che costituisce il cosiddetto "comitato" (*comittee*), anch'esso votato dagli *stakeholders*. Il comitato ha il potere di proporre delle modifiche ai parametri fondamentali del sistema, tra cui la dimensione dei blocchi, l'intervallo di tempo massimo richiesto per la creazione di un blocco, le ricompense per i testimoni. Quando una modifica proposta viene approvata dalla maggioranza dei delegati, l'intera comunità dei partecipanti alla rete ha la possibilità di invalidarla, votando per la deposizione del comitato.

Vi sono alcuni sistemi che non prevedono l'esistenza di un comitato. I poteri di sua competenza sono in questo caso assegnati ai testimoni.

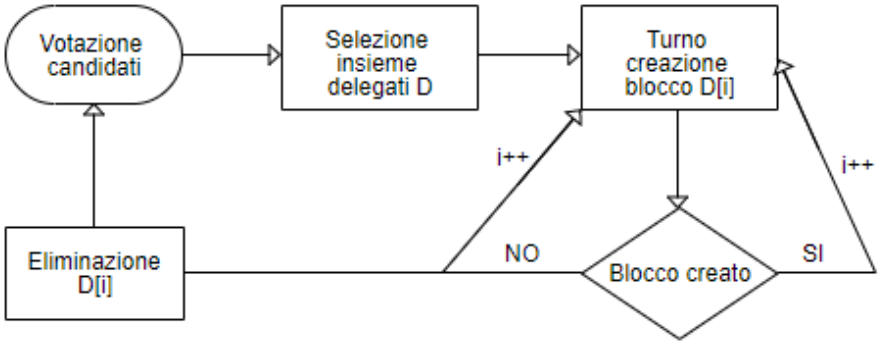

Figura 5.1: Flusso DPoS.

5.2 Considerazioni

Si è visto nel precedente capitolo come PoS cerchi di rimediare alle criticità di PoW, prima tra tutte quella relativa al consumo energetico. DPoS introduce un ulteriore risparmio sotto questo aspetto, dal momento che i consumi rilevanti sono quelli riguardanti i soli *N* nodi testimoni impegnati nel processo di creazione dei blocchi.
In DPoS la sicurezza della rete è strettamente legata al sistema di elezione dei testimoni. Tale sistema è impiegato infatti per risolvere i problemi relativi allo sbilanciamento di potere all'interno della rete: avere a disposizione una gran quantità di potenza computazionale in questo caso serve a poco, mentre possedere uno *stake* elevato consente unicamente di avere maggior peso nell'elezione dei delegati. Non vi è dunque un rapporto diretto tra le risorse ottenibili da un nodo e la sua probabilità di validare nuovi blocchi.
Esiste la possibilità che un nodo, o un gruppo di nodi, riesca ad ottenere la maggioranza dello *stake* disponibile nella rete cui partecipa, entrando in collusione con una minoranza di altri nodi per monopolizzare la creazione di blocchi. In tal caso, i candidati da esso selezionati nella votazione avranno un'alta probabilità di essere eletti testimoni e di mantenere la posizione permanentemente, arrivando a trasformare la democrazia di DPoS in una oligarchia.
Per ridurre questo rischio diverse piattaforme hanno adottato soluzioni differenti, tra cui [1, 8, 14]:

- Riduzione del numero dei testimoni;
- Introduzione di voti negativi: i partecipanti possono assegnare voti negativi a un candidato per ridurne la probabilità di elezione;
- Aumento della soglia di *stake* richiesto per raggiungere il quorum.

I suddetti accorgimenti, spesso utilizzati congiuntamente, contribuiscono a rendere gli attacchi più costosi e rischiosi e, conseguentemente, la rete più robusta.

Capitolo 6

Proof-of-Authority

Tra i meccanismi di consenso di più recente concezione si trova *Proof-of-Authority* (*PoA*), termine coniato da Gavin Wood, co-fondatore della blockchain *Ethereum*, in seguito alla prima implementazione di questo protocollo. Il suo scopo iniziale era la risoluzione degli attacchi che avevano portato la *testnet Ropsten* (catena alternativa basata su *Ethereum* e utilizzata per i test), che utilizzava PoW, ad uno stato inefficiente.
Oggi PoA è di particolare interesse, grazie alle sue prestazioni e alla tolleranza ai guasti, nelle blockchain *permissioned*, dove i partecipanti sono noti e affidabili.

6.1 Funzionamento

La strategia alla base di PoA può essere vista come una variante ottimizzata del metodo di consenso previsto da PoS, in cui, invece dello *stake*, ciò che ciascuno mette in gioco è la propria reputazione. Per partecipare come nodi attivi nel processo di validazione dei blocchi è infatti requisito fondamentale la verifica della propria identità.
Quando un partecipante vuole diventare validatore, si propone come tale certificando i propri dati anagrafici. Questo avviene in diversi modi a seconda dell'implementazione adottata [29], ad esempio può essere chi gestisce la blockchain ad occuparsi interamente della verifica affidandosi ad enti esterni, oppure le identità dei validatori possono essere registrate direttamente sulla blockchain, ad esempio tramite *smart contracts*, firmati con codici ottenuti mediante tradizionali meccanismi di *autenticazione multifattoriale.*
Così come avviene in DPoS, nella Proof-of-Authority esiste un gruppo ristretto di nodi abilitati alla validazione dei blocchi, detti *authorities*. Ad ogni *authority* è associato un *id* unico, tramite cui viene identificato il partecipante che agisce sulla blockchain.
Per mantenere il proprio status, le *authorities* sono tenute a preservare la sicurezza e l'efficienza della rete: se ad esempio tentasse di validare transazioni fraudolente, oppure

involontariamente non riuscisse a garantire prestazioni ottimali nel suo lavoro di validazione, il partecipante perderebbe i propri privilegi e la sua reputazione ne verrebbe danneggiata.

A titolo di esempio si prende ora in esame una implementazione di PoA analizzata nel dettaglio in [11]: *Authority Round* (*AuRa*) [2], adottata da *Parity*, un client *Ethereum*.

PoA adotta uno schema a rotazione per assegnare l'incarico di validazione alle *authorities* in carica nel sistema. Il tempo viene diviso in passi (*steps*), in ognuno dei quali viene selezionato un leader tra l'elenco dei validatori.

In AuRa si suppone che tutti i validatori siano sincronizzati con lo stesso tempo *t* (considerato convenzionalmente come il numero di secondi trascorsi a partire dalla mezzanotte del 1° gennaio 1970) e ognuno di essi calcola l'indice *s* di ogni passo con la formula:

$$s = t/D_s \tag{6.1}$$

dove D_s è una costante che indica la durata di ogni passo.

Ogni nodo *authority* ha due code: Q_{tx} e Q_b. Nella prima si raccolgono le transazioni trasmesse sulla rete, nella seconda vengono ricevuti i blocchi creati dagli altri validatori e in attesa di essere confermati.

Ad ogni passo viene eletto leader il validatore che ha identificativo dato da:

$$id = s \ mod \ N \tag{6.2}$$

dove *N* è il numero delle *authorities*.

Il leader *i* del passo s_i crea il suo blocco B_i con le transazioni reperite dalla sua coda Q_{tx} e lo propaga agli altri validatori, i quali comunicano tra loro trasmettendosi il blocco appena ricevuto. Se questi convengono nell'affermare di aver ricevuto il medesimo blocco, ognuno inserisce B_i nella propria coda Q_b.

Quando nel proprio turno un leader trova la coda Q_{tx} vuota, provvede ad inviare un blocco vuoto, poiché il protocollo prevede in ogni caso l'invio di un blocco. Se ciò non accade il leader viene ritenuto corrotto e gli altri validatori possono avviare una votazione per espellerlo dal gruppo di *authorities*. L'espulsione avviene quando almeno la maggioranza dei validatori (ovvero $\frac{N}{2} + 1$) vota a favore.

Se un nodo viene rimosso dalla sua posizione di *authority*, tutti i blocchi da esso creati e ancora presenti nelle code Q_b dei rimanenti validatori vengono ritenuti non validi e scartati. I blocchi che rimangono nelle code sono inseriti nella blockchain nel momento in cui la maggioranza delle *authorities* ha proposto il proprio blocco seguendo la procedura appena descritta.

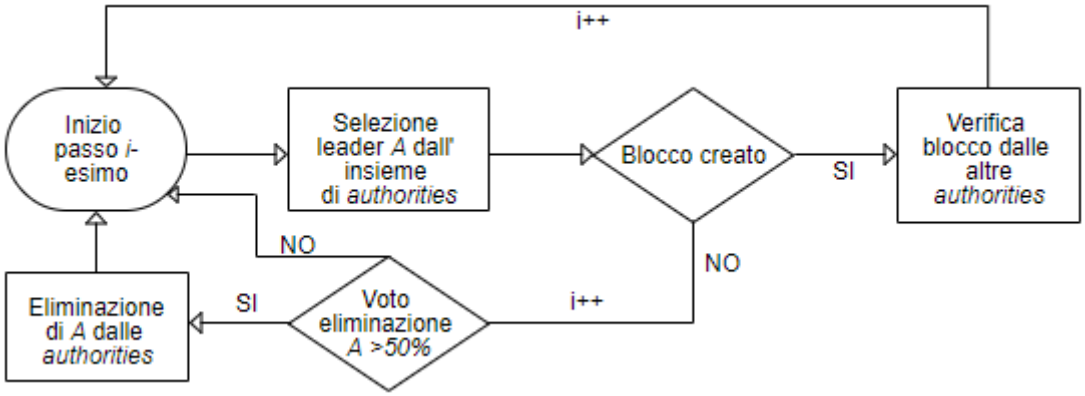

Figura 6.1: Flusso PoA AuRa.

6.2 Considerazioni

Si è visto come PoA richieda una fase di verifica dell'identità per i nodi candidati alla partecipazione attiva nel raggiungimento del consenso sulla blockchain. Questi sono in numero limitato rispetto alla totalità dei partecipanti al sistema e la fiducia degli altri nodi nelle loro decisioni è implicita proprio nella conoscenza della loro identità. Ciò richiede che il protocollo si appoggi a una qualche forma di autorità, interna o esterna al sistema, cui affidarsi per l'accertamento dei profili dei validatori. Questi aspetti comportano una diminuzione del grado di decentralizzazione. Per la sua natura dunque PoA si adatta bene non tanto al panorama delle blockchain pubbliche, quanto a quello delle soluzioni *permissioned*.

Per quanto riguarda l'aspetto dei consumi energetici, il design di PoA lo rende il protocollo più parsimonioso tra quelli analizzati, poiché non richiede una comunicazione diretta tra tutti i nodi della rete. Questa caratteristica lo rende un meccanismo efficiente anche dal punto di vista della velocità.

Grazie alle proprietà analizzate, PoA riesce a garantire un elevato livello di sicurezza. Ad esempio, un *51% attack* con un consenso di questo tipo è molto difficile da effettuare, dal

momento che un eventuale attaccante dovrebbe riuscire ad acquisire il controllo della maggioranza dei nodi validatori.

Si può concludere in definitiva che un approccio al consenso basato su PoA rappresenta un compromesso tra una scarsa decentralizzazione e una buona efficienza.

Capitolo 7

Analisi comparativa

Nei precedenti capitoli si è visto come funzionano alcuni dei protocolli per il consenso distribuito ad oggi esistenti, insieme ai rispettivi benefici, limiti e rischi per la sicurezza.
In questa sezione si valutano le strategie fin qui considerate in funzione delle proprietà fondamentali del teorema CAP (conosciuto anche come teorema di Brewer) [7], per poi confrontarne le prestazioni.

Teorema CAP

Il teorema CAP (*Consistency, Availability, Partition tolerance*) afferma che in un sistema informatico distribuito possono essere garantite contemporaneamente solo due delle seguenti tre proprietà [7, 11]:

- Consistenza: lo stato del sistema visibile ad un nodo è lo stesso visibile a tutti gli altri nodi. In una blockchain tale proprietà è anche definita *finalizzazione di consenso* e si dice che si ha consistenza quando è possibile evitare il formarsi di *fork*. Quando non è possibile raggiungere una perfetta consistenza, è necessario determinare se una *fork* viene prima o poi risolta (finalizzazione probabilistica) oppure no (nessuna finalizzazione).
- Disponibilità: ogni richiesta riceve una risposta. Nel contesto delle blockchain la disponibilità garantisce che una transazione propagata da un nodo sia considerata nella creazione dei blocchi e quindi, prima o poi, venga aggiunta alla catena.
- Tolleranza di partizione: il funzionamento del sistema non viene interrotto da eventuali errori o perdite di dati. Per le blockchain questo significa che il sistema continua a funzionare correttamente anche quando la comunicazione con alcuni nodi risulta interrotta.

Inoltre, in presenza di partizione è impossibile raggiungere allo stesso tempo consistenza e disponibilità. In una blockchain, poiché è necessaria la tolleranza di partizione, sono dunque possibili solo le combinazioni consistenza-tolleranza e disponibilità-tolleranza.
Nelle blockchain basate sulle strategie di consenso analizzate è possibile che vengano a crearsi delle ramificazioni nella catena. Tali algoritmi consentono cioè al sistema di entrare in uno stato di inconsistenza temporanea, risolvendo la situazione in modi differenti. Ad esempio, in PoW e PoS i nodi "onesti", dopo le opportune verifiche, considerano valida la catena più lunga e quelle alternative vengono abbandonate.
PoW, PoS, DPoS e PoA tendono a preferire la *disponibilità* a sfavore della *consistenza*, che è garantita unicamente come finalizzazione probabilistica.

Prestazioni

In una rete basata sulla tecnologia blockchain la valutazione delle prestazioni passa per i concetti di *throughput delle transazioni* e *latenza delle transazioni*. Il primo parametro misura la quantità di transazioni che il sistema è in grado di processare nell'unità di tempo. Il secondo parametro misura la quantità di tempo necessaria ad ogni singola transazione per essere effettivamente aggiunta alla blockchain, ovvero il tempo richiesto per la conferma del blocco contenente la transazione.
Sia throughput che latenza sono strettamente legati a variabili che dipendono dalla configurazione specifica del sistema e, di conseguenza, dal protocollo adottato per il raggiungimento del consenso, per questo è possibile che vi siano differenze notevoli anche tra algoritmi della stessa famiglia.
Il throughput è espresso in transazioni al secondo ed è così calcolato [27]:

$$Tx/s = \frac{B_s}{Tx_s * B_t} \tag{7.1}$$

dove B_s è la dimensione del singolo blocco, Tx_s è la dimensione di una transazione e B_t è il tempo medio richiesto per aggiungere un nuovo blocco alla catena. Il valore di B_t e l'insieme delle regole introdotte dal meccanismo di consenso determinano la latenza di una transazione. Ad esempio, se una rete prevede che, in presenza di una ramificazione, venga considerata valida la catena più lunga, il processo di verifica e selezione della versione corretta

della blockchain può rallentare la finalizzazione di un blocco, aumentando la latenza media delle transazioni.

In una rete basata su PoW, si prende qui ad esempio Bitcoin, si ha B_s = 1 *MB*, Tx_s = 250 *bytes* e B_t = 600*s*, perciò il throughput di transazione risulta di 7 *Tx/s*, che è un valore troppo basso per la maggior parte delle possibili applicazioni della tecnologia. Inoltre, nella rete Bitcoin un blocco è considerato finalizzato solo dopo la conferma di altri 6 blocchi, ciò significa che una transazione ha una latenza media di 6 * B_t = 3600*s*, ovvero deve attendere mediamente un'ora per essere finalizzata.

Per quanto riguarda i sistemi che adottano modelli alternativi a PoW, ciò che costituisce il collo di bottiglia per il raggiungimento di buoni risultati in termini di tempo è la comunicazione tra i nodi, piuttosto che lo svolgimento di un lavoro computazionale.

In PoS si hanno generalmente dimensioni dei blocchi superiori e tempi per la conferma dei blocchi inferiori rispetto a PoW [27]. Questo rende possibili valori più alti di throughput. Ad esempio, il protocollo Algorand arriva a processare le transazioni ad un ritmo di 875 *Tx/s* [18]. Anche sotto il profilo della latenza PoS è più performante di PoW e in alcuni sistemi che adottano questa strategia le transazioni vengono confermate in tempi nell'ordine dei secondi [23].

In DPoS la limitazione del numero di nodi validatori velocizza notevolmente la conferma dei blocchi. Ad esempio, la già citata blockchain BitShares ha dimostrato in uno stress test di poter raggiungere un throughput di 3300 *Tx/s* e una latenza di appena 1 secondo [5].

Come in DPoS, anche in PoA la presenza di un numero ristretto di validatori permette di ridurre lo scambio di messaggi tra i partecipanti alla rete e, di conseguenza, apre la strada ad un miglioramento delle prestazioni del sistema. Come si è visto nell'analisi di questa strategia, i blocchi vengono generati con una frequenza nota e controllata dal protocollo, perciò la velocità delle operazioni della blockchain dipende in larga misura dalla specifica configurazione utilizzata nelle diverse implementazioni dell'algoritmo. Un vantaggio di PoA è dato dalla possibilità di modificare, tramite il comitato, i parametri fondamentali del sistema, consentendo di ottimizzare dinamicamente le prestazioni.

Unitamente ai concetti appena visti di throughput e latenza viene spesso nominato quello di *scalabilità*, che può assumere sfumature differenti. In generale si dice che un sistema gode di

una buona scalabilità quando è in grado di espandersi senza modifiche strutturali e senza subire un degrado nelle prestazioni.

Le proprietà di PoW lo rendono una soluzione con scarsa scalabilità. Ad ogni generazione di un nuovo blocco è necessaria una sincronizzazione di tutti i nodi della rete, che deve avvenire prima della propagazione del blocco successivo. Ciò significa che il ciclo di creazione dei blocchi deve terminare in un tempo molto superiore a quello richiesto alla rete per sincronizzarsi, altrimenti verrebbero a formarsi numerose diramazioni della catena, che causerebbero ulteriori rallentamenti. Per questo PoW risulta lento e poco scalabile.

In PoS il meccanismo del consenso si basa su una selezione casuale del leader e non richiede lunghe computazioni. In questo caso il tempo richiesto per la sincronizzazione può essere paragonabile al tempo richiesto per il raggiungimento del consenso.

Il design di PoS rende possibile un aumento della dimensione dei blocchi e una riduzione dell'intervallo di tempo tra un blocco e il successivo. Questo si traduce in maggiore capacità di processazione della rete e buoni livelli di scalabilità. Lo stesso discorso vale per i modelli basati su DPoS e PoA.

L'ultimo aspetto che si valuta ora è quello riguardante la tolleranza ai guasti delle strategie analizzate, ovvero la capacità del sistema di fornire il proprio servizio anche in presenza di nodi guasti o malintenzionati. Come si è già visto, in PoW è necessario entrare in possesso della maggioranza della potenza computazionale per generare una catena alternativa, che possa imporsi come valida.

In PoS e DPoS un attaccante deve ottenere la maggioranza dello *stake* disponibile per minacciare il corretto funzionamento del sistema.

In PoA il raggiungimento del consenso dipende dalla presenza di una maggioranza di nodi onesti, dunque, detto N il totale dei partecipanti, la strategia è in grado di resistere ad un numero di nodi difettosi (o disonesti) pari a $N/2$.

Quanto fin qui detto viene sintetizzato nella seguente tabella:

PROTOCOLLO	POW	POS	DPOS	POA
CONSUMI ENERGETICI	Elevati	Bassi	Bassi	Trascurabili
PROPRIETÀ CAP	Disponibilità-Tolleranza	Disponibilità-Tolleranza	Disponibilità-Tolleranza	Disponibilità-Tolleranza
SCALABILITÀ	Bassa	Media	Media	Alta
TOLLERANZA GUASTI	50%	50%	50%	50%

Conclusioni

In questa tesi sono stati presi in considerazione i protocolli che regolamentano le operazioni e la gestione topologica di una rete blockchain. Una corretta progettazione di questi protocolli è di estrema importanza per poter garantire un funzionamento del sistema che sia il più robusto possibile.

Nello specifico, è stata esaminata la procedura per il raggiungimento del consenso distribuito nei meccanismi PoW, PoS, DPoS e PoA. Per ciascuno sono stati analizzati i punti di forza e le vulnerabilità sfruttabili da potenziali attaccanti, cercando di mostrare dove possibile le soluzioni note ai principali tipi di attacco.

Si sono infine paragonati i protocolli in termini di consistenza, disponibilità, throughput, latenza, scalabilità e tolleranza ai guasti. È risultato che tutti i modelli considerati sacrificano una perfetta consistenza in favore della disponibilità del sistema.

Si è evidenziato quanto la strategia PoW sia carente dal punto di vista dell'efficienza energetica e della scalabilità, mentre gli altri protocolli, nelle loro diverse implementazioni, consentono di raggiungere una migliore scalabilità. Questo li rende più adatti a molteplici applicazioni nel contesto delle reti distribuite.

Bibliografia

[1] Ark.io, “Ark Ecosystem Whitepaper”, Ark Whitepaper, 2019. Online: https://ark.io/Whitepaper.pdf (Ultimo accesso: 19/11/2019).

[2] AuRa. Online: https://wiki.parity.io/Aura (Ultimo accesso: 21/11/2019).

[3] S. Bano, A. Sonnino, M. Al-Bassam, S. Azouvi, P. McCorry, S. Meiklejohn, and G. Danezis, “Consensus in the age of blockchains”, In: *CoRR*, Vol. abs/1711.03936, 2017.

[4] G. Baxendale, ‘‘Can blockchain revolutionise EPRs?’’, In: *ITNOW*, Vol. 58, No. 1, pp. 38–39, March 2016.

[5] Bitshares Blockchain Foundation. “The BitShares Blockchain.”, Revised Whitepaper, 2018. Online: https://github.com/bitsharesfoundation/bitshares.foundation/blob/master/download/articles/BitSharesBlockchain.pdf (Ultimo accesso: 18/11/2019).

[6] A. Bondi, “Characteristics of scalability and their impact on performance.” In: *2nd International* Workshop *on Software and Performance — WOSP 2000*, ACM Press, Ottawa, Ontario, Canada, pp. 195–203, 2000.

[7] E. Brewer, “CAP twelve years later: How the ‘rules’ have changed.” In: *Computer*, Vol. 45, No. 2, pp. 23-29, 2012.

[8] G. Chaumont, P.Bugnot, Z. Hildreth, B. Giraux, “DPoPS: Delegated Proof-of-Private-Stake, a DPoS implementation under X-Cash, a Monero based hybrid-privacy coin.” *X-Cash Yellowpaper*, 2019. Online: https://x-network.io/whitepaper/XCASH_Yellowpaper_DPoPS.pdf (Ultimo accesso: 19/11/2019).

[9] K. Christidis, M. Devetsikiotis, "Blockchains and smart contracts for the Internet of Things", In: *IEEE Access*, vol. 4, pp. 2292-2303, 2016.

[10] P. Daian, R. Pass, E. Shi, "Snow white: Provably secure proofs of stake", *Cryptology ePrint Archive Report 2016/919*, 2016.

[11] S. De Angelis, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri, V. Sassone, "PBFT vs proof-of-authority: Applying the CAP theorem to permissioned blockchain", In: *Proc. ITASEC*, pp. 1-11, 2018.

[12] E. Deirmentzoglou, G. Papakyriakopoulos, and C. Patsakis, “A survey on long-range attacks for proof of stake protocols”, In: *IEEE Access*, Vol. 7, pp. 28 712–28 725, 2019.

[13] C. Dwork and M. Naor. “Pricing via processing or combatting junk mail.”, In: *12th Annual International Cryptology Conference*, pp. 139–147, 1992.

[14] EOS.io, “EOS.IO Technical White Paper v2”, EOS Whitepaper, 2018. Online: https://github.com/EOSIO/Documentation/blob/master/TechnicalWhitePaper.md (Ultimo accesso: 19/11/2019).

[15] R. Garavaglia. Tutto su Blockchain. Capire la tecnologia e le nuove opportunità. Hoepli, Milano, 2018. ISBN: 978-88-203-8314-5.

[16] P. Gazi, A. Kiayias, A. Russell, "Stake-Bleeding Attacks on Proof-of-Stake Blockchains", *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, pp. 85-92, 2018.

[17] A. Gervais, G.O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, S. Capkun, "On the security and performance of proof of work blockchains", In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security CCS ‘ 16*, New York, NY, USA:ACM, pp. 3-16, 2016.

[18] Y. Gilad, R. Hemo, S. Micali, G. Vlachos, N. Zeldovich, "Algorand: Scaling byzantine agreements for cryptocurrencies", In: *Proceedings of the 26th ACM Symp. Operating Syst. Principles*, pp. 51-68, 2017.

[19] S. Haber, W. S. Stornetta. “How to Time-Stamp a Digital Document.” In: *Journal of Cryptology*, Vol. 3, No. 2, pp. 99-111, 1991.

[20] M. Jakobsson and A. Juels. “Proofs of work and bread pudding protocols.”, In: *Proceedings of the IFIP TC6 and TC11 Joint Working Conference on Communications and Multimedia Security (CMS ’99)*, Leuven, Belgium, September 1999.

[21] A. Kiayias, A. Russell, B. David, R. Oliynykov, "Ouroboros: A provably secure proof-of-stake blockchain protocol", In: *Proceedings of the 37th Annu. Int. Cryptol. Conf. (CRYPTO)*, pp. 357-388, Aug. 2017.

[22] S. King, S. Nadal, "PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake", August 2012.

[23] J. Kwon, "TenderMint: Consensus without Mining", August 2014.

[24] R.C. Merkle. "One Way Functions and DES.", In: *Proceedings of Crypto '89*, pp. 428-446.

[25] McAfee,LLC. “Report sulle minacce alla blockchain”, 2018. Online: https://www.mcafee.com/enterprise/it-it/assets/reports/rp-blockchain-security-risks.pdf (Ultimo accesso: 17/11/2019).

[26] S. Nakamoto. “Bitcoin: A peer-to-peer electronic cash system” 2008. Online: https://bitcoin.org/bitcoin.pdf (Ultimo accesso: 10/11/2019).

[27] C. T. Nguyen, D. T. Hoang, D. N. Nguyen, D. Niyato, H. T. Nguyen, E. Dutkiewicz, "Proof-of-stake consensus mechanisms for future blockchain networks: Fundamentals applications and opportunities", In: *IEEE Access*, Vol. 7, pp. 85727-85745, 2019.

[28] Nxt Whitepaper. Online: https://web.archive.org/web/20150203012031/http://wiki.nxtcrypto.org/wiki/Whitepaper:Nxt#Blocks (Ultimo accesso: 17/11/2019).

[29] Online: https://blockchain4aid.org/consensus-mechanisms/proof-of-authority/ (Ultimo accesso: 20/11/2019).

[30] Peercoin documentation. Online: https://docs.peercoin.net/#/consensus-algorithm (Ultimo accesso: 17/11/2019).

[31] Post Bitcointalk. Online: https://bitcointalk.org/index.php?topic=27787.0 (Ultimo accesso: 17/11/2019).

[32] J. Siim. “Proof-of-Stake”, *Research Seminar in Cryptography*, 2017.

[33] W. Y. M. M. Thin, N. Dong, G. Bai, and J. S. Dong, “Formal analysis of a proof-of-stake blockchain,” In: *23rd International Conference on Engineering of Complex Computer Systems, ICECCS 2018*, Melbourne, Australia, December 12-14, 2018. IEEE, pp. 197–200, 2018.

[34] I. Weber, V. Gramoli, A. Ponomarev ; M. Staples, R. Holz, A.B. Tran ; P. Rimba, “On Availability for Blockchain-Based Systems.” In: *2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS)*, 2017.

[35] S. Zhang and J.-H. Lee, Analysis of the main consensus protocols of blockchain, ICT Express (2019). Online: https://doi.org/10.1016/j.icte.2019.08.001

[36] N. Szabo, Smart Contracts, 1994. Online: http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html.