Uno Sguardo Dentro Bitcoin

Giancarlo Giuffra Moncayo

23 maggio 2018

In questo articolo guarderemo dentro i meccanismi che permettono a Bitcoin di funzionare come mezzo per scambiare del valore. Inizieremo introducendo Bitcoin come un protocollo che risolve il problema del *double spending*. Parleremo poi di come Bitcoin offre una soluzione decentralizzata a tale problema grazie all'algoritmo di *Proof of Work*. Infine vedremo i dettagli di questo algoritmo di consenso.

Che cos'è Bitcoin?

Bitcoin[1] è un protocollo, cioè delle regole per costruire, interpretare e scambiarsi dei messaggi. Nel caso di Bitcoin questi messaggi rappresentano delle transazioni. Il protocollo[2] ha la caratteristica di essere peer-to-peer, cioè tutti i nodi che decidono di partecipare al protocollo hanno gli stessi privilegi, tutti i nodi sono uguali. D'altra parte, Bitcoin è anche l'oggetto di questi messaggi, cioè si chiama Bitcoin la crittovaluta o asset digitale che viene scambiato nelle transazioni. Anche se risulta facile capire dal contesto a quale Bitcoin uno si riferisce penso che sia utile fare attenzione alla differenza.

Essendo Bitcoin un protocollo, ovvero una specifica, ha diverse implementazioni. Queste implementazioni si chiamano bitcoin client[3], e sono il software che i nodi che formano parte della rete di Bitcoin hanno in esecuzione nei loro dispositivi, e.g. PC, GPU, ASIC. Il primo bitcoin client fu pubblicato nel 2009 da Satoshi Nakamoto. Questo client open source si è evoluto nel tempo e attualmente viene identificato con il nome di Bitcoin Core[4]. Il client viene mantenuto da una communità di sviluppatori che viene denominata con lo stesso nome del client.

Il Problema: Double Spending

Bitcoin nasce come protocollo per dare una soluzione peer-to-peer al problema del double spending, dove l'enfasi è in peer-to-peer. Questo problema consiste nell'impedire che un nodo del sistema possa spendere più di una volta la stessa moneta elettronica. Attualmente, quando effettuiamo dei pagamenti elettronici, il compito di controllare che non succedano casi di double spending viene delegato al circuito di pagamento utilizzato, che in modo centralizzato svolge tale compito.Tentativi di sistemi di pagamento elettronico[5] precedenti a Bitcoin risolvevano sempre in modo centralizzato questo problema. In effetti, la decentralizzazione di Bitcoin è una delle sue caratteristiche più innovative e importanti.

---

[1]Satoshi Nakamoto, il creatore di Bitcoin, pubblicò la prima specifica del protocollo in questo paper [1] del 2008.

[2]Per entrare nei dettagli dell'attuale protocollo si può partire da questa pagina [2] della bitcoin wiki.

[3]Si può trovare una lista dei maggiori bitcoin client qui [3].

[4]La pagina github di Bitcoin Core si può trovare qui [4].

[5]Trovate un resoconto dei predecessori di Bitcoin nella prefazione del libro Bitcoin and Cryptocurrency Technologies [5].

La Soluzione: Proof of Work

Bitcoin risolve il problema del double spending utilizzando l'algoritmo di Proof of Work. L'obbiettivo di questo algoritmo è quello di ordinare temporalmente le transazioni. In questo modo se ci sono due transazioni che cercano di spendere gli stessi Bitcoin la rete considererà valida solo la transazione che in accordo con l'algoritmo di Proof of Work è avvenuta per prima. L'innovazione in Bitcoin è che il consenso per quanto riguarda l'ordine temporale delle transazioni viene raggiunto in modo decentralizzato, ciascun nodo arriva alle stesse conclusioni in modo autonomo. Di fatto ciascun nodo ricostruisce l'intera catena di blocchi nel proprio dispositivo.

Come funziona la Proof of Work?

Per entrare nei particolari di questo processo penso che sia necessario introdurre alcuni concetti: transazioni, blocchi e il ruolo dei Miner. Partiamo dalla validazione delle transazioni.

Validazione delle Transazioni

Bitcoin è un protocollo per costruire, interpretare e scambiarsi transazioni. Ad ogni istante quindi ci sono nodi che inviano ai loro vicini le transazioni che hanno costruito secondo il formato del protocollo. Quando una di queste transazioni arriva a un nodo, il bitcoin client processa la transazione e determina se questa è valida o meno, e.g. controlla se i Bitcoin di questa transazione sono stati già spesi e che chi sta spendendo tali Bitcoin abbia effettivamente il diritto di farlo[6]. Se la transazione è valida il nodo propaga la stessa ai suoi vicini, altrimenti la transazione viene rifiutata e semplicemente non viene propagata alla rete. In questo modo tutti i nodi vengono a conoscenza delle transazioni valide che gli altri nodi creano.

I Miner

A questo punto entra in scena il ruolo del Miner, che raggruppa delle transazioni valide in una struttura dati particolare[7] e aggiunge alcuni altri dati. L'insieme di questi dati più la struttura che contiene le transazioni formano quello che viene chiamato un blocco, uno dei tanti che fanno parte

[6] Vedremo i dettagli di questa validazione in un successivo articolo, dopo aver studiato la struttura delle transazioni.

[7] La struttura per raggruppare le transazioni si chiama Merkle Tree [6], con un singolo hash può identificare un vasto gruppo di transazioni.

della blockchain. Un dettaglio importante è che tra i dati che il Miner aggiunge al blocco c'è l'hash del blocco precedente. Questo è il meccanismo che permette di legare temporalmente i blocchi formando così una catena che rappresenta il consenso della rete riguardo l'ordine delle transazioni (si veda Figura 1). La creazione di questi blocchi però è soggetta a certe regole. Abbiamo in effetti tralasciato un dettaglio importante, la validazione del blocco.

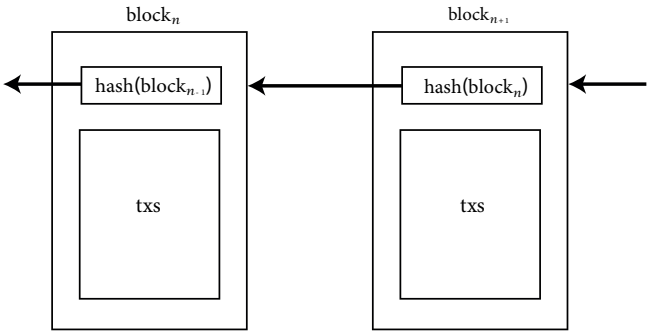


Figura 1: Ogni blocco contiene l'hash del blocco precedente. Questo meccanismo permette di legare temporalmente i blocchi.

Validazione dei Blocchi

Man mano che i blocchi vengono creati questi vengono propagati alla rete e come avete intuito vengono sottoposti ad una validazione da parte di ciascun nodo che li riceve, propagando solo quelli validi. In effetti una parte di tale validazione[8] è legata all'algoritmo di Proof of Work e riguarda l'hash del blocco. Un blocco è considerato valido solo se il suo hash è minore[9] di un certo valore chiamato *target*. Per soddisfare questa condizione il Miner ha a disposizione, tra i dati che aggiunge al blocco, una componente che può scegliere a suo piacimento. Questa componente viene chiamata *nonce*. La proprietà di non invertibilità della funzione di hash SHA-256 costringe il Miner a semplicemente provare diversi *nonce* per trovare un hash che soddisfi la condizione di validità. Non esiste effettivamente una strategia che garantisca una maggiore probabilità di trovare un hash valido[10]. L'unica

[8]Per avere una descrizione completa della validazione di un blocco si può vedere la sezione corrispondente nella seguente pagina [7].

[9]Siamo abituati a rappresentare un hash in base esadecimale, ma essendo un array di byte può facilmente rappresentarsi in base decimale ed ereditare così le operazioni di confronto che conosciamo.

[10]Si veda la sezione 3 del paper di hashcash [8].

altra variabile che il Miner può controllare oltre al *nonce* è la potenza di calcolo che decide di dedicare e che aumenta in modo proporzionale la sua probabilità di trovare un blocco valido. In questo modo il Miner inviando un blocco valido alla rete sta effettivamente dando prova di aver fatto del lavoro per mantenere sicura la rete, cioè è a tutti gli effetti una Proof of Work.

Calcolo del *target*

Rimane in sospeso il dettaglio di come viene determinato il valore di *target* che definisce la condizione di validità dell'hash di un blocco. Il protocollo Bitcoin utilizza questo *target* per mantenere attorno ai 10 minuti la frequenza di creazione dei blocchi. In effetti all'aumentare della capacità computazionale che i Miner dedicano alla rete, aumenta la probabilità che qualcuno trovi un hash valido e di conseguenza anche la frequenza di creazione dei blocchi. Vale ovviamente anche il viceversa. La scelta dei 10 minuti è stata fatta per motivi di stabilità e latenza[11]. Per adeguarsi quindi alla capacità computazionale della rete, il *target* viene aggiornato ogni 2016 blocchi. Ogni nodo della rete calcola l'intervallo temporale che è stato necessario per la creazione degli ultimi 2016 blocchi e lo compara con un intervallo di due settimane, cioè il tempo che sarebbe trascorso se tutti i blocchi fossero stati creati esattamente ogni 10 minuti. Dopodiché calcola la differenza percentuale tra queste due quantità e aggiorna l'attuale valore di *target* in base a tale percentuale[12](Si veda Figura 2).

Alcuni Dettagli Importanti

Abbiamo visto finora come i nodi Miner dedicano capacità computazionale alla rete per mantenere ordinate temporalmente le transazioni in una catena di blocchi e come tutti gli altri nodi, inclusi quindi anche eventuali nodi Miner, validano questo lavoro computazionale espresso in ciascun blocco. Abbiamo anche visto come la rete Bitcoin si autoregola per mantenere la capacità di processare transazioni stabile attorno ai 10 minuti per blocco. Vorrei adesso parlare di due dettagli importanti: chi sono effettivamente questi nodi Miner e come si risolve l'eventualità di ricevere due blocchi diversi ma validi che puntano allo stesso blocco precedente.

[11]Si veda la relativa FAQ [9] della bitcoin wiki per approfondire il tema della scelta dei 10 minuti.

[12]Si può trovare uno storico della difficoltà(inversamente proporzionale al *target*) qui [10].

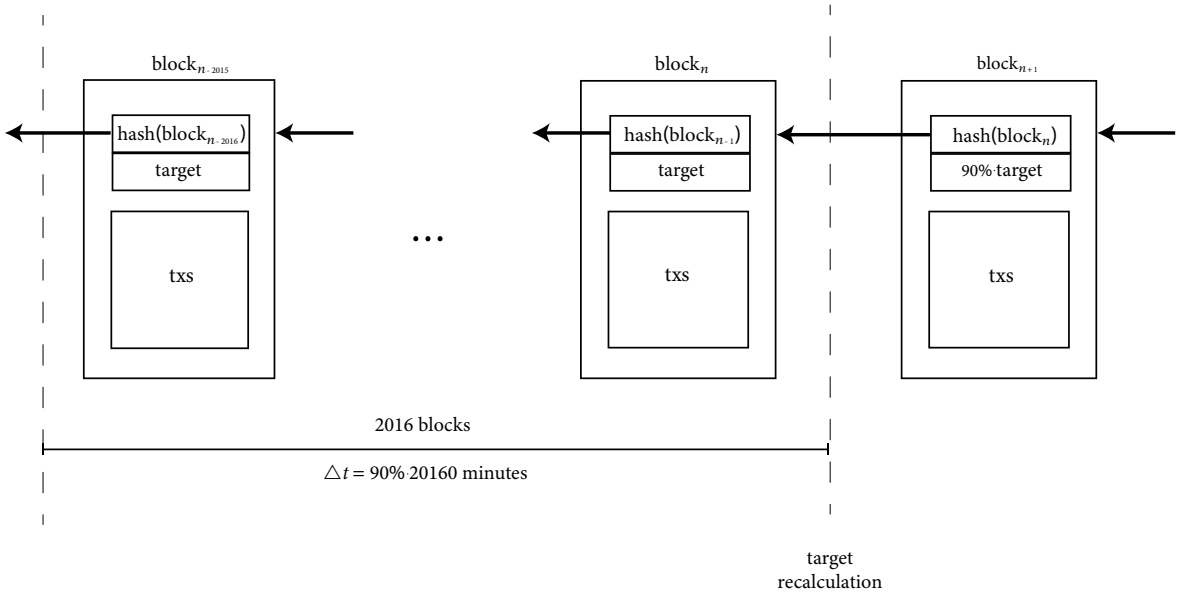


Figura 2: Ogni nodo calcola il tempo che la rete ha impiegato in creare gli ultimi 2016 blocchi e lo compara con un intervallo di 20160 minuti. Il *target* viene aggiornato secondo la differenza percentuale tra queste due quantità.

Chi sono effettivamente questi Miner?

La risposta si trova nella natura peer-to-peer di Bitcoin. Qualunque nodo può fare il Miner se ha delle risorse computazionali da dedicare alla rete. Ovviamente le risorse computazionali hanno un costo e questo è stato considerato nel disegno del protocollo. In effetti, il Miner ha diritto ad includere nel blocco una transazione addizionale[13]. Il beneficiario di questa transazione è scelto dal Miner e molto probabilmente sarà lui stesso. Invece l'ammontare della transazione è definito dal protocollo. Quindi il ruolo del Miner non è solo quello di mantenere sicura la rete ma anche quello di generare o minare Bitcoin. In questo modo il Miner ha un incentivo per dedicare risorse alla sicurezza della rete. Oltre a questa transazione, il Miner ha anche diritto a prendersi le commissioni[14] espresse in ciascuna transazione inclusa nel blocco che ha generato. Come abbiamo detto la quantità di nuovi Bitcoin che il Miner può mettere in circolazione dipende dal protocollo. Inizialmente questa quantità era 50 Bitcoin ma viene dimezzata ogni 210 000 blocchi, che corrispondono circa a 4 anni. Questo fa di Bitcoin una moneta

[13]Questa transazione viene chiamata coinbase [11].

[14]Le commissioni sono definite autonomamente dal nodo che ha generato la transazione. Vedremo come vengono specificate queste commissioni in un successivo articolo, dopo aver studiato la struttura di una transazione.

non inflazionaria, in effetti il meccanismo di dimezzamento garantisce che non potranno mai esistere più di 21 millioni di Bitcoin.

Come si risolvono i Fork?

Risulta possibile che due Miner in contemporanea trovino blocchi validi. Questi due blocchi saranno propagati nella rete e di conseguenza i nodi avranno due catene valide che differiscono unicamente nell'ultimo blocco, cioè ci saranno due fork[15]. Questa ambiguità temporanea viene risolta quando il blocco successivo viene trovato. Questo nuovo blocco sarà in effetti collegato solo a uno dei due blocchi e il bitcoin client sceglierà, come da protocollo, la catena che abbia la difficoltà complessiva più alta consentendo alla rete di raggiungere così di nuovo il consenso. Il valore di difficoltà viene calcolato in base al *target*. Quindi normalmente la catena con più difficoltà complessiva sarà quella più lunga. Solo in concomitanza con l'aggiornamento del valore di *target* è probabile che questo non sia il caso (Si veda Figura 3).

Conclusione

Questo sguardo dentro Bitcoin ci è servito per capire come si incastrano i diversi pezzi del protocollo e quali sono le diverse fasi della vita di una transazione (Si veda Figura 4). Abbiamo però parlato in modo generico del concetto di transazione e non abbiamo descritto come vengono effettivamente trasferiti i Bitcoin. Nell'articolo successivo analizzeremo la struttura di una transazione.

Bibliografia

[1] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

[2] Bitcoin Wiki. Protocol documentation. https://en.bitcoin.it/wiki/Protocol_documentation.

[3] Bitcoin Wiki. Clients. https://en.bitcoin.it/wiki/Clients.

[4] Bitcoin Core. Github. https://github.com/bitcoin/bitcoin.

[15] Questo tipo di fork è temporaneo e previsto dal protocollo. Non è da confondersi con i concetti di Hard Fork e Soft Fork, che sono due metodologie diverse per aggiornare un'applicazione distribuita com'è il caso di un bitcoin client.

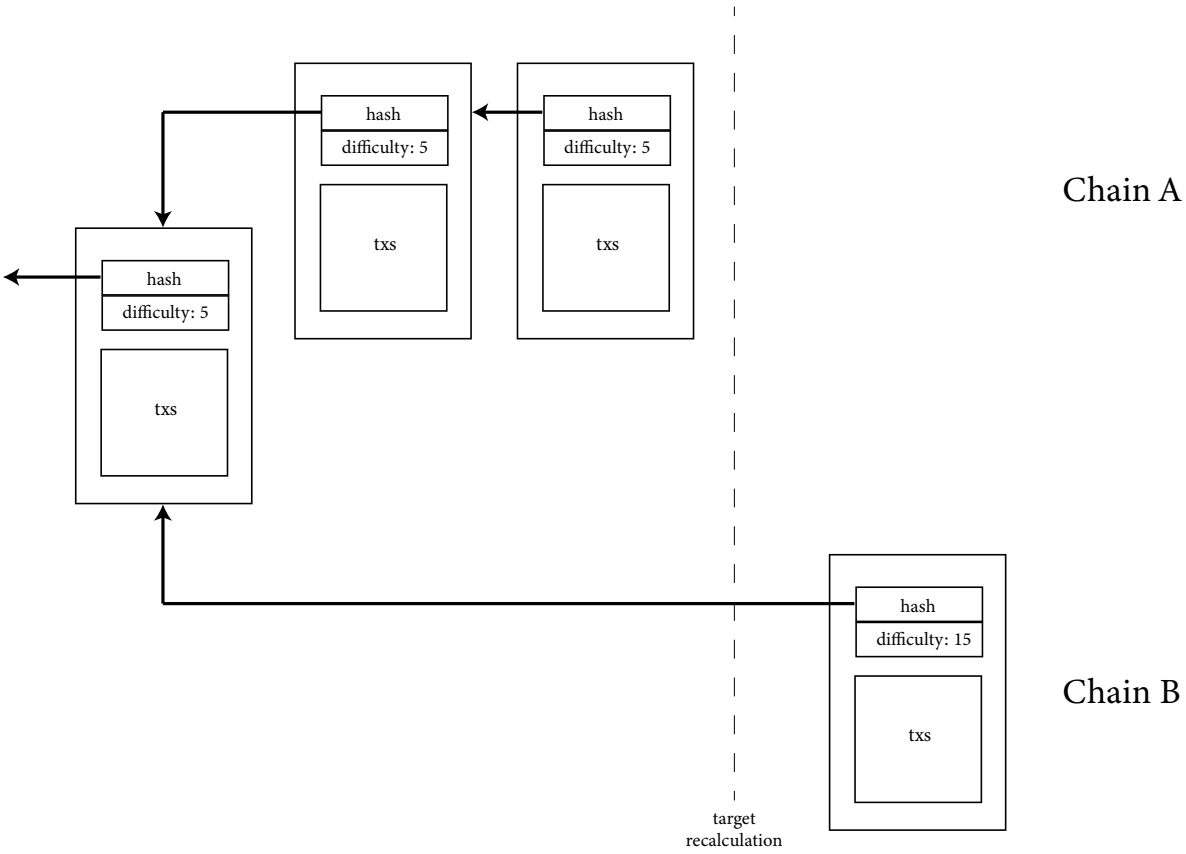


Figura 3: Si osserva come in concomitanza di un aggiornamento di *target* è possibile che la catena con la difficoltà complessiva più alta non sia quella più lunga. Nell'esempio la catena B, più corta di un blocco rispetto alla catena A, ha una difficoltà complessiva maggiore.

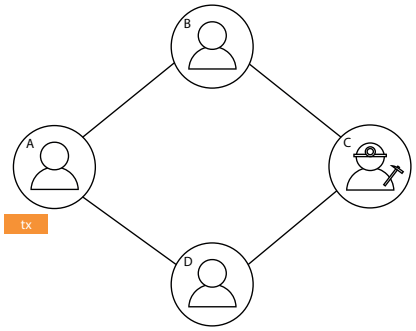
[5] Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton University Press, 2016.

[6] Wikipedia. Merkle tree. https://en.wikipedia.org/wiki/Merkle_tree.

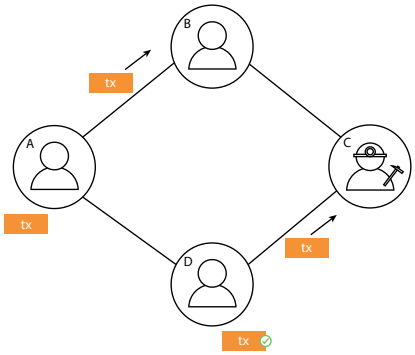
[7] Bitcoin Wiki. Protocol rules. https://en.bitcoin.it/wiki/Protocol_rules.

[8] Adam Back. Hashcash-a denial of service counter-measure. 2002.

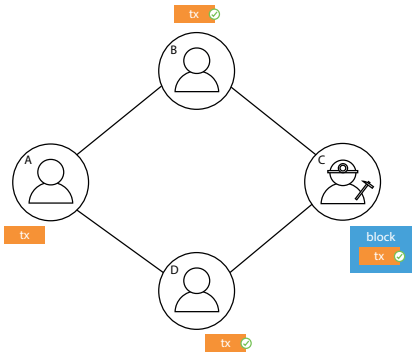
[9] Bitcoin Wiki. Faq. https://en.bitcoin.it/wiki/Help:FAQ.



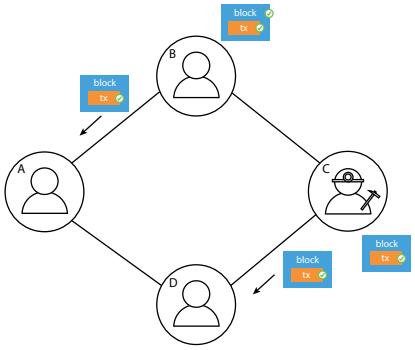
(a) Il nodo A crea una transazione.



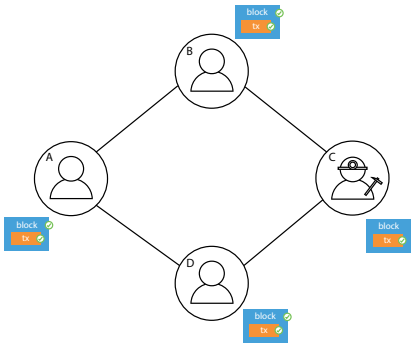
(b) Il nodo A invia la transazione ai suoi vicini. Il nodo D l'ha ricevuta, validata e propagata.



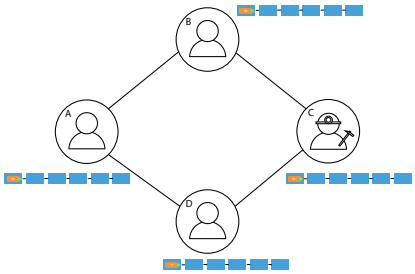
(c) Il Miner C valida la transazione, la include in un blocco e genera una Proof of Work valida.



(d) Il Miner C invia il blocco ai suoi vicini. Il nodo B l'ha ricevuto, validato e propagato.



(e) Il nodo A riceve il blocco che contiene la sua transazione. Lo valida e così verifica la sua prima *confirmation*.



(f) Il nodo A aspetta fino a validare la sesta *confirmation* per considerare la transazione definitiva.

Figura 4: Nello schema si vede lo stato di una rete Bitcoin semplificata ad ogni fase della vita di una transazione: generazione, propagazione e validazione, inclusione in un blocco, propagazione e validazione del blocco, e infine la prima *confirmation*. Si ricorda che si consiglia di aspettare fino ad avere 6 *confirmation* per considerare la transazione definitiva.

[10] BitcoinWisdom. Difficulty. https://bitcoinwisdom.com/bitcoin/difficulty.

[11] Bitcoin Wiki. Coinbase. https://en.bitcoin.it/wiki/Coinbase.