"METODO IMPLEMENTATO PER MEZZO DI UN COMPUTER PER RAGGIUNGERE UN CONSENSO DISTRIBUITO IN UNA RETE BLOCKCHAIN E NODO CHE IMPLEMENTA IL METODO"

DESCRIZIONE

CAMPO DELLA TECNICA

La presente invenzione riguarda un metodo implementato per mezzo di computer per raggiungere un consenso distribuito basato su puntate per un registro distribuito, come la tecnologia blockchain.

STATO DELLA TECNICA

L'invenzione è particolarmente adatta, ma non limitata, a raggiungere un livello più alto di distribuzione di una rete basata su puntate, al fine di non avere troppe puntate concentrate in troppo pochi nodi, aumentando così la solidità della rete e riducendo la sua vulnerabilità a qualsiasi nodo che viene compromesso.

In questo documento viene impiegato il termine blockchain per comprendere tutte le forme di registri elettronici, basati su computer e distribuiti. Questi comprendono, pur senza esserne limitati, tecnologie blockchain e di catene di transazioni, registri autorizzati e non autorizzati, registri basati sul consenso, registri condivisi e loro variazioni. L'applicazione più nota della tecnologia blockchain è il registro Bitcoin, sebbene siano state proposte e sviluppate altre implementazioni blockchain. Un blockchain è un registro elettronico basato sul consenso, il quale viene implementato come un sistema decentrato e distribuito basato su computer, costituito da blocchi i quali a loro volta sono costituiti da transazioni ed altre informazioni. Bitcoin è un esempio di blockchain con prova di lavoro in cui operatori eseguono calcoli costosi per facilitare transazioni sul blockchain. I blockchain basati su prove di lavoro sono stati criticati a causa delle grandi risorse di calcolo richieste, il che richiede un grande consumo di energia per funzionare ed anche

per il fatto che la generazione di blocchi può essere irregolare e lenta. Inoltre, diversi blocchi devono essere costruiti al di sopra di un determinato blocco prima che il dato blocco sia considerato confermato (ovvero, è sufficientemente improbabile che venga ripristinato).

I blockchain basati su proof of stake sono stati proposti in alternativa a blockchain con prova di lavoro. In una rete blockchain con proof of stake, il blockchain è protetto da proof of stake anziché da prova di lavoro. In una proof of stake, gli operatori detengono una puntata (depositando alcuni token) su un conto speciale. Questa puntata può essere definita deposito cauzionale e la probabilità di essere selezionata come il nodo per estrarre un blocco è proporzionale alla quantità delle risorse digitali fornite come deposito cauzionale. Le reti blockchain con proof of stake possono essere utilizzate per evitare le spese di calcolo e l'energia necessaria per un mining su blockchain con prova di lavoro. Inoltre, le blockchain con proof of stake possono consentire una frequenza più elevata ed una creazione di blocchi più regolare rispetto a blockchain con prova di lavoro. Almeno alcune blockchain con proof of stake hanno anche una bassa probabilità di biforcazione ed un blocco può essere efficacemente confermato non appena viene aggiunto alla blockchain.

Come accennato in precedenza, sebbene l'economia di scala porti alla concentrazione di risorse di mining per aumentare l'efficienza della rete, nello sviluppo di una rete distribuita, la concentrazione è un problema relativo al controllo della rete, alla robustezza della rete poiché è impossibile eseguire operazioni di manutenzione pianificata sui nodi. Il problema principale nelle reti piccole e concentrate riguarda la bassa ridondanza in caso di guasti, siano essi problemi di rete, hardware o software. La situazione opposta, in cui vi è un'eccessiva frammentazione, può anche essere controproducente. Questo perché la trasmissione e la replica delle informazioni presentano sempre un sovraccarico all'interno di un sistema reale. Inoltre, l'assegnazione di puntate direttamente a nodi di mining può comportare problemi di sovraccarico se sono presenti vincoli di distribuzione.

Pertanto, è necessario affrontare i problemi di concentrazione sopra citati e migliorare il protocollo di consenso con proof of stake per ottenere un protocollo che incoraggi la distribuzione ed allo stesso tempo l'efficienza della rete.

SCOPI DELL'INVENZIONE

Uno scopo della presente invenzione, secondo un primo dei suoi aspetti, è quello di ottenere un metodo implementato da computer in grado di incoraggiare la distribuzione della rete e l'efficienza degli minatori, e nello stesso tempo ottimizzare la scalabilità della blockchain.

Un secondo scopo della presente invenzione è quello di ottenere un metodo implementato da computer in grado di migliorare la robustezza della rete.

Un ulteriore scopo è quello ottimizzare il metodo implementato da computer, al fine di aumentare la sicurezza globale della rete blockchain.

Un ulteriore scopo della presente invenzione è quello di fornire un metodo implementato da computer il quale incoraggi l'impiego di nodi con prestazioni oneste per estrarre il blocco successivo e di scoraggiare l'impiego di nodi i quali non funzionano bene o hanno comportamenti intenzionalmente cattivi o dannosi.

Uno scopo della presente invenzione è anche quello di mantenere basso il consumo di energia della rete e possibilmente aumentare la potenza di calcolo del nodo solo quando è necessario.

Un ulteriore scopo della presente invenzione è quello di fornire un sistema il quale implementa il metodo implementato da computer in modo facilmente progettabile, con un consumo di energia e di calcolo ottimizzato ed estremamente efficiente.

BREVE DESCRIZIONE DELL'INVENZIONE

Nel seguito vengono riassunti alcuni aspetti tecnici della presente invenzione, i quali consentono di raggiungere alcuni degli scopi più importanti.

Secondo un primo aspetto, questa invenzione si riferisce ad un metodo implementato da computer per raggiungere un consenso distribuito basato su proof of stake per una rete

blockchain, permettendo così di raggiungere un livello più alto di distribuzione della rete e di non avere troppe puntate concentrati in troppo pochi nodi, detta rete blockchain comprendendo uno o più indirizzi MAIN, per ciascun indirizzo MAIN essendo associati a loro volta uno o più indirizzi/nodi OVERFLOW, l'ultimo essendo in grado di fungere da potenziali nodi di mining, solo se viene assegnato un quantitativo minimo prestabilito di puntate a detti uno o più indirizzi/nodi di OVERFLOW, comprendendo i seguenti processi:

- un processo di ridistribuzione configurato per calcolare la puntata minima necessaria per un nodo OVERFLOW da estrarre, essendo minimum_stake_value = total_amount_at stake/predetermined_target_number_of_nodes, per calcolare la puntata massima che consente un mining efficiente, essendo maximum_stake_value = minimum_stake_value*2, per ciascun indirizzo MAIN per distribuire la puntata tra i nodi OVERFLOW, per scorrere i nodi OVERFLOW di un dato indirizzo MAIN e per assegnare a ciascun nodo OVERFLOW il minimum_stake_value, fino a quando la rimanente puntata non è sufficiente per attivare un miner su ogni nodo overflow che è stato assegnato alla sua puntata;

-un processo di penalità configurato per congelare una quantità di premio corrispondente al lavoro computazionale per aver minato un numero di slot oltre un limite prestabilito Slot_MAX in un periodo, e per recuperarlo nei periodi seguenti a partire dal $periodo_{i+1}$, un po' alla volta in base ad un numero di parti proporzionale al numero di slot minati su detto limite prestabilito Slot_MAX, essendo una parte assegnata a ciascuno slot successivo che il miner può venire delegato ad estrarre, questo al fine di incentivare naturalmente la distribuzione della rete blockchain.

Un tale metodo implementato da computer consente naturalmente un'alta distribuzione della rete blockchain.

In base ad un secondo aspetto, la presente invenzione si riferisce ad un metodo implementato da computer, nel quale il processo di ridistribuzione può essere ulteriormente configurato in modo tale per cui, se rimane una qualsiasi puntata, questa viene distribuita tra i nodi OVERFLOW fino a quando il maggior numero possibile di nodi ha una puntata uguale al valore di puntata massimo e ciascuna ulteriore puntata viene assegnata al primo nodo OVERFLOW secondo un ordine prestabilito.

Esso permette di ottimizzare la ridistribuzione della quota totale assegnata ad un pool di nodi OVERFLOW collegati ad un indirizzo MAIN e fa in modo che solo primi nodi OVERFLOW in base ad una numerazione prestabilita progressiva univoca possono trovarsi sotto un regime di penalità

Secondo un terzo aspetto, la presente invenzione riguarda un metodo implementato da computer comprendente inoltre un processo di inizializzazione configurato per elencare tutti gli indirizzi MAIN della rete blockchain, per ciascun elenco di indirizzi MAIN essendo assegnati indirizzi/nodi OVERFLOW, e quindi per ciascun indirizzo OVERFLOW/nodo trasferisce la quota, se ne possiede una, al suo indirizzo MAIN assegnato.

Esso permette di ottimizzare il processo di ridistribuzione nel caso in cui vi possa essere una qualsiasi quota su un nodo OVERFLOW prima dell'implementazione del processo di distribuzione.

Secondo un quarto aspetto, la presente invenzione riguarda un metodo implementato da computer il quale comprende inoltre un processo di manutenzione, comprendente le seguenti operazioni, alle quali viene assegnato un indirizzo MAIN ed almeno due o più indirizzi OVERFLOW ed un nodo che si desidera manutenere:

110: invio di una transazione comprendente un primo input che fornisce un indirizzo OVERFLOW sostitutivo utilizzando il suo codice privato, il che significa che detto indirizzo OVERFLOW sostitutivo è un nodo che sostituisce il nodo che si desidera manutenere

120: assegnazione di detto indirizzo OVERFLOW sostitutivo al corrispondente indirizzo MAIN utilizzando il codice privato dell'indirizzo MAIN

130: annullamento della registrazione desiderata da manutenere utilizzando il suo codice privato. In alternativa, l'indirizzo MAIN può ANNULLARE L'ASSEGNAZIONE di un nodo di overflow precedentemente assegnato ad esso.

Esso permette di manutenere la rete blockchain altamente distribuita anche in caso di guasto di uno o più nodi di mining.

Secondo ulteriori aspetti, la seguente invenzione riguarda:

- un nodo configurato per implementare un metodo per raggiungere un consenso distribuito basato su proof of stake per una rete blockchain, permettendo così di raggiungere un livello più alto di distribuzione della rete e di non avere troppe puntate concentrate in troppo pochi nodi, detto nodo comprendendo un dispositivo di interfaccia, un processore accoppiato a detto dispositivo di interfaccia, una memoria accoppiata al processore, la memoria avendo immagazzinate su di essa istruzioni eseguibili mediante computer le quali, quando eseguite, configurano il processore per eseguire le corrispondenti operazioni del metodo implementato dal computer descritto.

- un mezzo di memorizzazione leggibile da computer comprendente istruzioni eseguibili da computer le quali, una volta eseguite, configurano un processore per eseguire il metodo implementato dal computer descritto.

- una rete di computer per eseguire una transazione basata su blockchain da un primo utente a un secondo utente in una rete di trasferimento elettronica di nodi collegati, in cui i nodi comprendono almeno un primo nodo, un secondo nodo descritto, in cui ciascun nodo condivide una stessa blockchain, ed in cui ciascun nodo memorizza un mezzo di memorizzazione leggibile da computer descritto.

Secondo un aspetto della presente invenzione, si propone uno schema completamente decentralizzato di determinazione dell'ordine di accesso ad una risorsa condivisa. L'idea di base è quella di ordinare i nodi, e.g. di OVERFLOW, abilitati all'accesso alla risorsa

condivisa secondo un proprio identificativo. A ciascun nodo abilitato viene attribuito un intervallo numerico di ampiezza proporzionale al numero degli slot temporali in cui lo stesso nodo ha diritto di avere accesso alla risorsa condivisa.

Un nodo risulta abilitato all'accesso quando ha diritto ad accedere alla risorsa condivisa per almeno un time slot.

Conseguentemente, gli intervalli numerici sono consecutivi ed anch'essi ordinati secondo il medesimo ordine degli identificativi univoci dei nodi.

Tale allineamento di intervalli numerici è condiviso e conosciuto da tutti i nodi. Nel tempo, tale allineamento può cambiare in relazione alla presenza ed ampiezza dei singoli intervalli numerici.

Per ogni epoca i nodi abilitati all'accesso alla risorsa condivisa calcolano od acquisiscono un seme, vale a dire un numero casuale o pseudocasuale. Per epoca si intende, in sintesi secondo l'approccio TDMA - "Time division multiple access", un macro intervallo di suddivisione del tempo, in cui ciascuna epoca è suddivisa in cosiddetti time slot. Il numero dei time slot attribuiti a ciascun nodo che condivide una risorsa condivisa è predeterminato secondo una predeterminata logica di distribuzione.

Ciascun nodo esegue in modo autonomo le seguenti operazioni in successione per un numero di cicli pari al numero di time slot dell'epoca:

- i-mo calcolo dell' i-mo Hash del seme,
- Calcolo del resto della divisione di detto i-mo Hash per la somma delle ampiezze di detti intervalli numerici,
- Determinazione dell'intervallo in cui cade detto resto,
- Attribuzione dell'i-mo slot al nodo corrispondente a detto intervallo.

La rete può essere formata da un numero maggiore rispetto ai nodi abilitati. La redistribuzione dei time slot tra i nodi può dipendere da politiche autoritative oppure da politiche casuali.

Il seme è un numero casuale o pseudo-causale che può essere estratto da uno specifico nodo che per esempio ha il solo compito di generare il seme.

Quando la presente invenzione è applicata all'ambito delle blockchain, la risorsa condivisa consiste nel compito di minare un blocco ed il seme può essere determinato sulla base di un hash costruito sulla base di uno o più blocchi di un'epoca precedente.

Ad esempio si può prevedere di ripartire il numero di blocchi minati di un'epoca precedente in tre o quattro gruppi e di considerare l'hash del primo blocco di ciascun gruppo oppure l'hash del secondo blocco di ciascun gruppo, oppure il primo, secondo e terzo hash del secondo gruppo oppure l'hash dell'ultimo, penultimo e terzultimo blocco del primo insieme di blocchi.

Gli hash estratti in questo modo sono concatenati. Il loro concatenamento oppure l'hash del loro concatenamento (hash di hash) definisce il seme per il calcolo dell'ordine di accesso alla risorsa condivisa della prossima epoca.

Vantaggiosamente, dal momento che tutti i nodi detengono una copia della blockchain, ciascun nodo è in grado di calcolare l'ordine in cui ciascun nodo abilitato ha diritto di accedere alla risorsa condivisa.

Gli elementi essenziali che ciascun nodo deve conoscere è, la ripartizione dei time slot tra i nodi abilitati, l'identificativo univoco di tutti i nodi abilitati ed il seme.

Secondo l'applicazione della presente variante realizzativa alle blockchain, queste sono del tipo Proof of Stake, in cui non esiste una competizione per completare la costruzione o mining dei blocchi. Gli slot sono assegnati ai nodi secondo il metodo descritto sopra e ciascun nodo ha il compito di completare un blocco all'interno di tale time slot, inserendo transazioni accodate e pendenti al momento in cui il time slot ha inizio.

Vantaggiosamente, la presente forma di realizzazione consente ad una blockchain di proseguire nei propri compiti anche in caso di casuali e momentanee disconnessione dei nodi.

Le performance di un nodo si possono apprezzare in relazione al numero di transazioni che questo riesce ad includere in un blocco generato nel time slot assegnato.

Al termine della generazione del blocco, il nodo assegnatario del time slot, lo inoltra alla rete di nodi, i quali accettano o respingono tale blocco, se risultato corretto. Quando il blocco è accettato dal 50%+1 dei nodi, il blocco è considerato come definitivamente accettato e pertanto, la blockchain può continuare a crescere accodando ulteriori blocchi a tale blocco.

Secondo una variante preferita, una volta trascorso il time slot, se il nodo assegnatario non invia per tempo agli altri nodi il blocco generato nello stesso time slot, pertanto denominato "blocco tardivo", il nodo assegnatario del time slot successivo inizia a generare un proprio blocco, denominato "nuovo blocco", ignorando la mancata o tardiva ricezione del blocco tardivo. Questo implica che alcune delle transazioni presenti nel blocco tardivo possono essere incluse nel nuovo blocco. Si determina pertanto una biforcazione ed una competizione tra il blocco tardivo ed il nuovo blocco. La competizione è vinta dal blocco che per primo riceve il 50%+1 di approvazione da parte dai restanti nodi della rete. Pertanto, l'accettazione del blocco tardivamente inoltrato o del nuovo può dipendere dall'entità del ritardo.

Un ulteriore blocco viene dunque accodato al blocco che vince la contesa. Bisogna tenere conto che, nelle blockchain, il blocco che viene accodato include l'hash del blocco a cui è accodato.

BREVE DESCRIZIONE DELLE FIGURE

Le caratteristiche strutturali e funzionali della presente invenzione ed i suoi vantaggi rispetto alla tecnica precedente nota, diventeranno ancora più chiari in base alle rivendicazioni che seguono, ed in particolare da un esame della seguente descrizione, effettuata con riferimento alle figure allegate le quali mostrano una forma di realizzazione

schematica preferita ma non limitata del metodo, sistema, dispositivo implementato da computer secondo l'invenzione, in cui:

la figura 1 illustra uno schema di flusso del metodo implementato da computer secondo la presente invenzione;

La figura 2 illustra un esempio operativo del processo di registrazione del metodo implementato da computer secondo la presente invenzione;

La figura 3 illustra uno schema di un esempio della rete comprendente un indirizzo MAIN e tre nodi OVERFLOW secondo la presente invenzione;

la figura 4 illustra uno schema di flusso del processo di inizializzazione del metodo implementato da computer secondo la presente invenzione;

la figura 5 illustra un diagramma di flusso operativo del processo di inizializzazione del metodo implementato da computer secondo la presente invenzione;

la Figura 6 illustra un esempio operativo del processo di inizializzazione del metodo implementato da computer secondo la presente invenzione;

la figura 7 illustra un diagramma di flusso del processo di ridistribuzione del metodo implementato da computer secondo la presente invenzione;

la figura 8 illustra un diagramma di flusso operativo del processo di ridistribuzione del metodo implementato da computer secondo la presente invenzione;

la figura 8 illustra un diagramma di flusso operativo del processo di ridistribuzione del metodo implementato da computer secondo la presente invenzione;

la figura 9 illustra un esempio operativo del processo di ridistribuzione del metodo implementato da computer secondo la presente invenzione;

la figura 10 illustra uno schema di flusso del processo di penalità del metodo implementato da computer secondo la presente invenzione.

la figura 11 è una tabella di associazione di entità necessarie al calcolo autonomo di accesso ad una risorsa condivisa;

nella figura 12 è mostrato un intervallo numerico;

nella figura 13 è mostrato uno schema di accesso ad una risorsa condivisa secondo una forma di realizzazione preferita della presente invenzione;

nella figura 14 è mostrato un diagramma di flusso di un metodo secondo una forma di realizzazione preferita della presente invenzione.

Nell'ambito della presente descrizione il termine "secondo" componente non implica la presenza di un "primo" componente. Tali termini sono infatti adoperati come etichette per migliorare la chiarezza e non vanno intesi in modo limitativo.

DESCRIZIONE DETTAGLIATA DELL'INVENZIONE

In generale, questa divulgazione descrive un metodo implementato da computer, un sistema per raggiungere un consenso distribuito basato su proof of stake per un registro distribuito, come la tecnologia blockchain. In particolare, la presente invenzione riguarda una tecnologia blockchain basata su proof of stake, nella quale possessori di puntate non possono e non devono comunicare con nodi di mining.

La tecnologia blockchain secondo la presente invenzione prevede due attori principali: possessori di puntate, ossia cioè quelli che possiedono token che consentono il controllo della catena, ed minatori, cioè quelli che controllano nodi di mining attivamente operanti nella rete ed incaricati di un sufficiente ammontare di puntate per ottenere l'incarico di un certo numero di slot di lavoro, dove questi estraggono un corrispondente numero determinato di blocchi allungando così la catena.

La blockchain è considerata un meta-attore ed è l'unione dei nodi che operano sulla rete. L'invio di una transazione alla blockchain comporta che questa transazione raggiunga ciascun nodo operativo che aggiorna costantemente il suo stato interno. Queste procedure di aggiornamento sono costruite in modo tale da garantire che le stesse condizioni iniziali producano identici aggiornamenti di stato in ciascun nodo in modo strettamente deterministico. Questo meta-attore rappresenta la massiccia procedura di aggiornamento su ciascun nodo della rete.

D'ora in poi si intende che:

- uno slot è un intervallo di tempo di 30 secondi. Questi slot sono assegnati in modo univoco ai nodi di mining ed all'interno di un determinato slot uno ed un solo nodo ha il diritto di produrre un blocco. Questa assegnazione è unica e strettamente deterministica e viene eseguita prima dell'inizio di ciascun periodo;
- per un periodo si intende un raggruppamento di slot contigui. Ciascun periodo è costituito da un numero uguale di slot. L'assegnazione e la ridistribuzione di slot ed azioni sono calcolate una volta per ciascun periodo e rimangono invariate fino alla fine dello stesso.
- una puntata è una rappresentazione numerica del diritto di voto di ciascun possessore di puntate della rete. La puntata può essere delegata con regole ed operazioni specifiche ai nodi coinvolti nella creazione di un blocco.
- indirizzo MAIN: questo indirizzo non è associato ad un oggetto fisico (un indirizzo MAIN non è associato ad un nodo di mining attivo) ed agisce come un segnaposto il quale indica un gruppo di risorse fisiche al quale si desidera che la blockchain possa accedere, esponendo così il suo codice privato solo durante una transazione di assegnazione dell'indirizzo OVERFLOW all'indirizzo MAIN e non durante un'altra transazione operativa, aumentando la sicurezza globale della catena; questo indirizzo MAIN (ad esempio alice_main_address) è dichiarato dal suo proprietario e viene gestito tramite il suo codice privato (ad esempio alice_key). Solo l'indirizzo MAIN può dichiarare quali nodi OVERFLOW gli sono assegnati;
- indirizzo OVERFLOW: questo indirizzo è associato ad un oggetto fisico e funge da nodo di mining attivo; questo indirizzo OVERFLOW (ad esempio node_1_overflow_address) viene gestito utilizzando il suo codice privato (ad esempio node_1_key); è possibile che il gestore di indirizzi MAIN ed il gestore di nodi OVERFLOW siano la stessa entità, però essi utilizzano codici differenti quando impersonano Alice o Node_1; il proprietario di ciascun indirizzo MAIN (ad esempio

Alice) utilizzando una transazione specifica associa ciascun nodo di overflow registrato al suo indirizzo MAIN. Solo l'indirizzo MAIN nella rete può collegare un indirizzo OVERFLOW a se stesso, il che significa che solo l'indirizzo MAIN può inviare una transazione di tipo ASSIGN la quale collega un indirizzo OVERFLOW ad un indirizzo MAIN. Questa operazione non è commutativa, ovvero un indirizzo OVERFLOW può non assegnare se stesso ad un indirizzo MAIN. L'indirizzo/nodo OVERFLOW può non venire condiviso tra indirizzi MAIN, un indirizzo OVERFLOW può venire assegnato ad un solo indirizzo MAIN. Solo un indirizzo precedentemente dichiarato come OVERFLOW può essere associato ad un indirizzo MAIN. Un nodo OVERFLOW può anche annullare la registrazione e, in tal modo rimuove se stesso sia dal pool di nodi disponibili sia dall'indirizzo MAIN al quale è stato precedentemente assegnato. Questo per dare ad entrambi i nodi MAIN e OVERFLOW la possibilità di proteggere gli stessi da situazioni indesiderate nelle quali essi vengono collegati tra loro, sia che ciò sia il risultato di un errore in buona fede, sia di un comportamento malintenzionato.

Se un indirizzo OVERFLOW non è collegato a nessun indirizzo MAIN, esso sarà operativo solo come un nodo di replica.

Il metodo implementato da computer secondo la presente invenzione (fig. 1) comprende un processo di ridistribuzione, un processo di penalità e può comprendere un processo di registrazione, un processo di inizializzazione ed un processo di manutenzione.

Il metodo implementato da computer secondo la presente invenzione è impostato secondo i seguenti parametri e valori, laddove detti valori devono essere intesi solo come esempi e possono venire modificati in base a specifiche esigenze tecniche:

Durata oraria dello slot = 30 secondi

Slot nel periodo (E_S) = 24000 => numero di slot per ciascun periodo

Fattore di penalità = 2 => determina la quantità di penalità di tempo per la violazione del protocollo corrente

Fattore di recupero = 1 => determina l'età del ritorno dal regime di penalità

Numero target di nodi (MAX_N) = 400 => numero ottimale di nodi che formano la rete

Numero minimo di nodi (min_N) = 200

Slot minimo per nodo Slot_min (E_S/MAX_N) = 60

Slot massimo per nodo Slot_MAX (E_S/min_N) = 120

Punto di valutazione di un periodo (EEP) (E_S/3) = 8000

Il processo di registrazione è configurato per registrare uno o più indirizzi MAIN, per registrare uno o più indirizzi/nodi OVERFLOW e per associare uno o più indirizzi/nodi OVERFLOW ad un indirizzo MAIN e per registrare tale associazione.

Il processo di registrazione (fig. 2, 3) implementato su ciascun nodo di overflow della rete blockchain comprende le seguenti operazioni:

5: tramite la rete blockchain, ricevere una transazione comprendente come input un indirizzo MAIN e registrarla in un registro di indirizzi utilizzando un codice privato del gestore dell'indirizzo MAIN

10: tramite la rete blockchain, ricevere una transazione comprendente come input uno o più indirizzi OVERFLOW e registrare uno o più indirizzi OVERFLOW nel registro di indirizzi utilizzando un codice privato del gestore di ciascun indirizzo OVERFLOW

15: tramite la rete blockchain, ricevere una transazione comprendente come input l'assegnazione di uno o più indirizzi OVERFLOW all'indirizzo MAIN registrato nell'operazione 5.

Il metodo implementato da computer secondo la presente invenzione prevede di impostare una quantità minima di puntate (S_min_VAL) assegnata ad un nodo OVERFLOW per venire attivata e considerata come nodo di mining ed al fine di evitare che tutte le puntate concentrate solo su pochi nodi vengano impostate, fissando un limite superiore corrispondente alla quantità di puntate che può venire assegnata ad un nodo, oltre la quale il nodo di mining inizia ad incorrere in penalità. Ciò dovrebbe rendere una concentrazione di puntate al di là di intervalli di efficienza meno allettante per nodi che

desiderano raccogliere i massimi premi possibili. La quantità massima di puntate desiderata (S_MAX_VAL) può essere il doppio del valore di S_min_VAL.

L'algoritmo è progettato per massimizzare il numero di nodi attivi diffondendo la puntata in eccesso tra il maggior numero possibile di nodi di overflow al fine di espandere la rete aumentando così la sua resilienza.

Il metodo implementato da computer comprende un processo di inizializzazione al fine di recuperare la puntata indirizzata in modo non corretto su un nodo OVERFLOW e di raccoglierle nel corrispondente indirizzo MAIN prima di eseguire un processo di ridistribuzione secondo il metodo implementato da computer dell'invenzione. Il processo di inizializzazione è utile quando puntate vengono indirizzate in modo non corretto su uno o più nodi OVERFLOW, rendendo così affidabile la distribuzione tra i nodi OVERFLOW. Inoltre, il processo di inizializzazione è utile per garantire che agli indirizzi OVERFLOW

non siano assegnate puntate prima di eseguire il processo di ridistribuzione.

Il processo di inizializzazione (fig. 4) è configurato per elencare tutti gli indirizzi MAIN della rete blockchain, per ciascun elenco di indirizzi MAIN i suoi indirizzi/nodi OVERFLOW assegnati, e quindi per ciascun indirizzo/nodo OVERFLOW trasferire l'azione, se ne ha, al suo indirizzo MAIN assegnato.

Il processo di inizializzazione (figg. 5, 6) implementato su ciascun nodo di overflow della rete blockchain in particolare comprende le seguenti operazioni:

20: avere una rete blockchain comprendente uno o più indirizzi MAIN, per ciascun indirizzo MAIN essendo associati a loro volta uno o più indirizzi/nodi di OVERFLOW i quali agiscono come potenziali nodi di mining solo se una quantità minima prestabilita di puntate è assegnata a detto uno o più indirizzo/nodo di OVERFLOW;

25: tramite la rete blockchain, ricevere una transazione comprendente come input indirizzi dal registro indirizzi che comprende indirizzi MAIN e loro corrispondenti indirizzi OVERFLOW;

30: associare a ciascun indirizzo MAIN un numero progressivo univoco;

35: per ciascun indirizzo MAIN verificare se esso contiene una quantità di puntate e se questo valore è> 0, assegnare questo valore al parametro M_A_S, e se questo valore è <= 0, eseguire l'operazione 35 fino al successivo elemento di indirizzo MAIN della rete blockchain e generare una transazione comprendente come output l'esecuzione del valore M_A_S associato al corrispondente indirizzo MAIN,

40: associare a ciascun nodo OVERFLOW un numero progressivo univoco

45: per ciascun nodo OVERFLOW assegnato a ciascun indirizzo MAIN corrispondente verificare se esso contiene una quantità di definita di puntate OVERFLOW_ADDRESS.assigned_stake e se questo valore è> 0, assegnare questo valore al parametro O_A_S, calcolando il valore MAIN_ADDRESS.assigned_stake come M_A_S + O_A_S ed assegnare O_A_S = 0 e generare una transazione comprendente come output il valore aggiornato del valore M_A_S associato al corrispondente indirizzo MAIN, quindi eseguire l'operazione 45 fino al successivo indirizzo OVERFLOW assegnato al corrispondente indirizzo MAIN considerato nell'operazione 35,

e se OVERFLOW_ADDRESS.assigned_stake è <= 0, effettuare l'operazione 45 fino al successivo indirizzo OVERFLOW assegnato al corrispondente indirizzo MAIN considerato nell'operazione 35,

50: generare una transazione comprendente come output il valore aggiornato del valore TOTAL_AT_STAKE per ciascun indirizzo MAIN della rete blockchain corrispondente alla quantità totale di puntate su ciascun indirizzo MAIN dopo aver eseguito le operazioni precedenti.

Il processo di ridistribuzione (fig. 7) è configurato per calcolare la puntata minima necessaria per estrarre un nodo OVERFLOW, essendo minimum_stake_value = total_amount_at stake/predetermined_target_number_of_nodes, per calcolare la puntata massima che consente un mining efficace, essendo il valore massimo di puntata = valore minimo di puntata * 2 , per ciascun indirizzo MAIN distribuendo la puntata tra i nodi

OVERFLOW, scorrendo attraverso i nodi OVERFLOW di un dato indirizzo MAIN ed assegnare a ciascun nodo OVERFLOW il valore minimum_stake_value, fino a quando la puntata rimanente non è sufficiente per attivare un miner o a ciascun nodo di overflow non è stata assegnata la sua puntata. Il processo di ridistribuzione può inoltre essere configurato in modo tale, per cui se rimane una puntata questa viene distribuita tra i nodi OVERFLOW fino a quando il maggior numero possibile di nodi ha una puntata pari al valore massimo della puntata e ciascuna altra puntata viene assegnata al primo nodo OVERFLOW secondo un ordine prestabilito.

Il processo di ridistribuzione (figg. 8, 9) comprende a sua volta un primo processo di ridistribuzione dell'iterazione in cui l'algoritmo assegna a ciascun nodo OVERFLOW disponibile assegnato a ciascun indirizzo MAIN la quantità minima di puntate necessaria per diventare un nodo di mining attivo, ed in un secondo processo di ridistribuzione di iterazione l'algoritmo distribuisce la puntata cercando di massimizzare il numero di nodi di overflow che operano in una puntata completa in vista di S_MAX_VAL impostato per qualsiasi nodo OVERFLOW.

Il processo di ridistribuzione comprende in particolare le seguenti operazioni:

52: avere una rete blockchain comprendente uno o più indirizzi MAIN, per ciascun indirizzo MAIN essendo associati a loro volta uno o più indirizzi/nodi di OVERFLOW i quali agiscono come potenziali nodi di mining solo se una quantità minima prestabilita di puntate è assegnata a detto uno o più indirizzi/nodi di OVERFLOW;

55: ottenere come input un valore TOTAL_AT_STAKE per ciascun indirizzo MAIN della rete blockchain corrispondente alla quantità totale di puntate su ciascun indirizzo MAIN

60: calcolare S_min_VAL = TOTAL_AT_STAKE/TARGET NUMBER OF_NODES essendo TARGET NUMBER OF_NODES = numero totale di nodi di mining OVERFLOW nella rete della catena,

Calcolare S_MAX_VAL = S_min_VAL * 2

Supponendo un valore di contatore ITERATION = 0

65: se il valore ITERATION è <2, aggiornare allora il valore di contatore ITERATION = ITERATION + 1

70: per ciascun indirizzo MAIN che ottiene come input il valore MAIN_ADDRESS.assigned_stake e se questo valore è> 0, il che significa che l'indirizzo MAIN in esame contiene una quantità di puntate, assegnare questo valore al parametro M_A_S,

e se il valore M_A_S è> = S_MAX_VAL ripetere l'operazione 70 fino al successivo indirizzo MAIN,

75: se il valore M_A_S è <S_MAX_VAL, ricavare come input il valore OVERFLOW_ADDRESS.assigned_stake per ciascun indirizzo OVERFLOW associato al MAIN in esame, e

se il valore OVERFLOW_ADDRESS.assigned_stake è = 0 ripetere l'operazione 75 fino al successivo indirizzo OVERFLOW associato al MAIN in esame e

se il valore OVERFLOW_ADDRESS.assigned_stake è ≠ 0, assegnare il valore OVERFLOW_ADDRESS.assigned_stake al parametro O_A_S, calcolando MAIN_ADDRESS.assigned_stake = M_A_S-S_min_VAL, e calcolando OVERFLOW_ADDRESS.assigner_stake = O_A_S + S_min_VAL, e ripetere l'operazione 75 fino al successivo indirizzo OVERFLOW associato al MAIN in esame.

Il processo di penalità rende svantaggiose le operazioni di mining che non rispettano le regole stabilite, incentivando così i nodi ad espandere naturalmente la rete aumentando così la sua capacità di recupero, come già trattato.

D'ora in poi si intende:

$periodo_i$, il periodo in cui gli minatori hanno registrato i loro indirizzi MAIN e OVERFLOW,

$periodo_{i+1}$, il periodo in cui gli minatori estraggono i loro slot assegnati,

$periodo_{i+2}$, il periodo in cui vengono valutati premi e penalità.

Il pedice "j" riguarda il $blocco_j$ scritto nello $slot_j$

Il pedice "i" riguarda il $periodo_i$ e ciò che è rilevante per il $periodo_i$.

Il metodo implementato da computer secondo la presente invenzione comprende un premio per gli minatori per la remunerazione del loro sforzo di calcolo durante la loro operazione di mining. Se un miner non è in grado di scrivere un blocco nello slot corrispondente o estrae in un numero di slot oltre un limite prestabilito, egli non riceverà nel momento successivo il premio corrispondente per il blocco non scritto o per il blocco minato oltre il limite prestabilito (in ciascuno slot può essere scritto solo un blocco), ma questo premio (per i blocchi minati oltre il limite) viene congelato e verrà recuperato molto lentamente un po' alla volta (ad esempio secondo un numero di parti pari al doppio del numero di slot minati in regime di penalità) per ciascuno slot successivo che sarà stato minato dal miner, in particolare in base a un numero di parti proporzionale al numero di slot minati oltre detto limite prestabilito, Slot_MAX, essendo una parte assegnata a ciascuno dei seguenti slot che miner può essere delegato ad estrarre, quindi al fine di incentivare naturalmente la distribuzione della rete blockchain.

Come menzionato prima, un $premio_{i+2}$ corrisponde al premio per estrarre un $blocco_{j+1}$ nel $periodo_{i+1}$.

Questo $premio_{i+2}$ comprende una base monetaria (ovvero una nuova moneta creata) per il $blocco_{,\,i+1}$ minato, ed ed una commissione di $raccolta_{j,i\,i+1}$. Questa commisssione di $raccolta_{j,\,i+1}$ a sua volta comprende la commissione raccolta pagata dagli utenti che desiderano eseguire una transazione e scritta nel relativo $blocco_{j,\,i+1}$ della rete blockchain .

D'ora in poi si intende:

$Penalty\_slots_{i+1}$ = numero di slot minati oltre il limite nel $periodo_{i+1}$ ovvero numero di slot su Slot_MAX * 2

$Commisssione\ congelata_{i+1}$ = è la quantità di premio non erogata agli minatori nel $periodo_{i+1}$ per aver minato un numero di slot oltre il limite Slot_MAX, essendo quindi stato soggetto ad un regime di penalità.

Il processo di penalità (fig. 10) implementato su ciascun nodo di overflow della rete blockchain comprende in particolare le seguenti operazioni:

80: calcolare la commissione congelata totale$_{i+1}$ quale importo del premio$_{i+2}$ non erogato agli minatori in un momento specifico$_{i+1}$ per aver minato un numero di slot oltre il limite Slot_MAX

85: calcolare i penalty_slots$_{i+1}$ preferibilmente per due volte (o secondo un'altra legge proporzionale) il numero di slot minati oltre il limite, cioè il numero di slot su Slot_MAX

90: calcolare recovered_fee$_{j,\ i+1}$ valore come commissione congelata$_{i+1}$/penalty_slots$_{i+1,}$ ovvero la commissione da recuperare per ogni slot

95: aggiungere il valore recovered_fee$_{j,\ i+1}$ al premio del MAIN_address per l'mining del blocco$_{i}$ nel periodo$_{i+1}$ come premio$_{j,\ i+1}$ = un coinbase + collect_fee$_{j,\ i+1}$ + recovered_fee$_{j,\ i+1}$

100: calcolare il nuovo valore del parametro frozen_fee$_{i+1}$ come frozen_fee$_{i+1}$ - recovered_fee$_{j,\ i+1}$ ed il nuovo valore di penalty_slots $_{i+1}$ come penalty_slots$_{ii+1}$ -1 e ripetere l'operazione 95 per il blocco seguente$_{j+1}$ nel periodo$_{i+1}$ mentre frozen_fee $_{i+1} > 0$ o mentre penalty_slots$_{i+2} > 0$.

Alla fine di ciascun periodo, corrispondente all'ultimo blocco minato, penalty_slots e frozen_fee vengono trasferiti dal primo nodo OVERFLOW al suo indirizzo MAIN. Allo stesso modo, all'inizio di ciascun periodo, corrispondente al primo blocco minato, i penalty_slots e frozen_fee vengono trasferiti dall'indirizzo MAIN al suo primo nodo OVERFLOW.

Questa procedura è necessaria per evitare comportamenti malintenzionati come trasferimenti indebiti di token da un pool ad un altro.

Inoltre, è sempre possibile registrare un nuovo nodo di overflow e calcolare il suo indirizzo in modo che preceda nell'ordine di overflow quello con le penalità assegnate. In tal modo, un pool recupererebbe le commissioni congelate senza reinserirsi nei parametri ottimali o estendere la rete. Considerando che in circostanze normali le commissioni

superano la base monetaria generata ciò ridurrebbe in modo significativo l'efficacia delle penalità.

Per quanto riguarda eventi di manutenzione programmata al fine di mantenere la rete blockchain altamente distribuita anche in caso di guasto di uno o più nodi di mining, l'invenzione implementata da computer secondo la presente invenzione può comprendere un processo di manutenzione comprendente a sua volta le seguenti operazioni, ricevendo un indirizzo MAIN ed almeno due o più indirizzi OVERFLOW ed un nodo che si desidera manutenere:

110: inviare di una transazione comprendente un primo input che fornisce un indirizzo OVERFLOW sostitutivo utilizzando il suo codice privato, il che significa che detto indirizzo OVERFLOW sostitutivo è un nodo che sostituisce il nodo che si desidera manutenere

120: assegnare detto indirizzo OVERFLOW sostitutivo al corrispondente indirizzo MAIN utilizzando il codice privato dell'indirizzo MAIN

130 annullare la registrazione del nodo che si desidera manutenere utilizzando il suo codice privato. In alternativa l'indirizzo MAIN può ANNULLARE L'ASSEGNAZIONE di un nodo di overflow precedentemente assegnato ad esso.

Per quanto riguarda aspetti di sicurezza della blockchain secondo la presente invenzione, secondo il modo con cui è strutturato il sistema di assegnazione, codici privati dell'indirizzo MAIN entrano in gioco solo per il breve momento di tempo necessario per generare la registrazione e la transazione di assegnazione, però non sono più coinvolti in alcuna successiva fase di elaborazione. Questo lo esenta dall'essere disponibile online su un oggetto che potrebbe essere soggetto ad attacchi malintenzionati. D'altro canto, una presenza costante online di un server fisico ed i codici corrispondenti necessari per operare sulla rete consentono ad una terza parte malintenzionata di rubare loro codici privati e di assumere tutte le azioni delegate all'insieme degli indirizzi MAIN.

La presente invenzione riguarda inoltre un nodo (preferibilmente un nodo OVERFLOW) configurato per implementare il metodo implementato da computer descritto, detto nodo comprendendo un dispositivo di interfaccia, un processore accoppiato al detto dispositivo di interfaccia, una memoria accoppiata al processore, la memoria avendo immagazzinato su di essa istruzioni eseguibili da un computer le quali, una volta eseguite, configurano il processore per eseguire le operazioni corrispondenti del metodo implementato dal computer descritto.

La presente invenzione riguarda inoltre un mezzo di memorizzazione leggibile da computer comprendente istruzioni eseguibili da computer le quali, quando vengono eseguite, configurano un processore per eseguire il metodo implementato da computer descritto.

La presente invenzione riguarda inoltre una rete di computer per eseguire una transazione basata su blockchain da un primo utente ad un secondo utente in una rete di trasferimento elettronico di nodi collegati, in cui i nodi comprendono almeno un primo nodo, un secondo nodo, come descritto in precedenza, in cui ciascun nodo condivide una stessa blockchain, ed in cui ciascun nodo memorizza un mezzo di memorizzazione leggibile da computer descritto in precedenza.

Nel seguito vengono descritti alcuni esempi operativi del metodo implementato da computer secondo la presente invenzione.

<u>Primo esempio: operatore malintenzionato</u>

Si suppongano due indirizzi MAIN:

- Alice, alice_main_addr, node_a1, node_a2, ..., node_a10 puntata 30.000
- Mallory, mallory_main_addr, node_m1, node_m2,…, puntata node_m5 30.000

Puntate totali: 1.200.000
Puntata minima: 3.000 Puntata massima: 6.000
Slot min: 60 Slot max: 120

Ad ogni blocco generato viene assegnato un premio di 1 token

Nel periodo 14 Alice e Mallory hanno completato la registrazione dei loro indirizzi principali ed hanno assegnato loro i loro nodi.

Questi saranno attivi a partire dal periodo 15

Nel periodo 16 viene calcolato il premio.

All'inizio del periodo 15 Alice avrà 10 nodi attivi (node_a1, ..., node_a10) ciascuno con 60 slot assegnati.

All'inizio del periodo 15, Mallory avrà 5 nodi attivi (node_m1, ..., node_m5) ciascuno con 120 slot assegnati.

Gli slot di Alice e Mallory sono gli stessi poiché hanno la stessa puntata attiva. Tuttavia, la distribuzione cambia. Mallory ha solo 5 nodi mentre Alice ne ha 10. Nessuno di questi è penalizzato. I nodi di Alice sono attivi con il minimo, i nodi di Mallory con il massimo.

Si supponga una perfetta esecuzione del protocollo nel periodo 15, senza transazioni per semplificare l'esempio.

I dieci nodi Alice hanno assegnato 60 slot, producendo così 60 blocchi; in ciascun blocco vi sarà un premio di 1 token. Essendo dieci i nodi (60 * 10 * 1) il premio sarà di 600 token. All'inizio del periodo 16, il saldo di Alice aumenterà da 30.000 a 30.600 token.

Passiamo ora a Mallory: i cinque nodi di Mallory hanno assegnato 120 slot, producendo così 120 blocchi; in ogni blocco vi sarà un premio di 1 token. Essendo cinque i nodi (120 * 5 * 1) il premio previsto sarà di 600 token. All'inizio del periodo 16, il saldo di Mallory dovrebbe aumentare da 30.000 a 30.600 token. Solo tre dei nodi Mallory, node_m3, node_m4, node_m5, sono stati modificati per funzionare in modo errato. Le cause di questo comportamento possono essere varie ed ora ne elencheremo alcune.

● È stata apportata una modifica intenzionale allo scopo di eseguire operazioni non valide (transazioni duplicate, accettazione di transazioni false con codice errato, ...) = Interazione malintenzionata volontaria.

● Il server su cui è installato il nodo non funziona correttamente e fallisce le operazioni matematiche contrassegnando come transazioni valide non valide ad esempio Carol, con

un budget di 100 token, e ne invia 25 a Bob. L'operazione è legale ma è contrassegnata come illegale nel blocco del prodotto . = Interazione malintenzionata involontaria.

- La connessione di rete del server non è affidabile ed il blocco non viene trasmesso in tempo. = Interazione malintenzionata involontaria.
- Il blocco non è collegato al suo predecessore valido immediato ma a un altro predecessore che viola la regola di estendere la catena più pesante. In questo modo, il nodo mira a far scomparire transazioni indesiderate, ad esempio un pagamento, dalla storia della catena = Interazione malintenzionata involontaria.

Questi aspetti fanno sì che i blocchi generati dal node_m5 vengano rifiutati dagli altri nodi e non diventino parte della blockchain.

In questo modo il premio raccolto da Mallory sarà:

Nodo	**Premio raccolto**
node_m1	120
node_m2	120
node_m3	0
node_m4	0
node_m5	0
Totale	240

Passiamo ora ad una valutazione della parte rimanente della rete.

Alice e Mallory hanno in totale (30.000 Alice e 30.000 Mallory) 60.000 token di puntate.

Sul resto della rete (1.200.000 - 60.000) rimangono in gioco 1.140.000 token.

Supponendo che tutto il resto della rete funzioni in condizioni ideali e senza transazioni con gli altri nodi che compongono la rete, verranno assegnati:

Slot totali	Slot di Alice	Slot di Mallory	Slot assegnati al

			resto della rete
24000	-600	-600	= 22800

Verranno generati 22.800 blocchi e quindi un premio di 22.800 token.

Puntata totale	Puntata di Alice	Puntata di Mallory	Puntata assegnata al resto della rete
1200000	-30000	-30000	= 1140000

Nel periodo 15 la puntata totale della rete è di 1.200.000, la puntata assegnata a nodi non Alice o Mallory è di 1.140.000. Nel periodo 16 il premio per i nodi sopra menzionati sarà di 22.800 token, portando la loro puntata a 1.162.800 token.

Considerando che il Mallory principale ha raccolto una puntata inferiore, abbiamo ricalcolato i valori per il periodo successivo.

La puntata complessiva è di 1.200.000 + 600 + 240 + 22.800 = 1.223.640

La tabella seguente confronta la percentuale di controllo della rete nel momento 15 con la rivalutazione nel periodo 16 (prevista) e 16 (reale)

	Periodo 15	**Periodo 16 (previsto)**	**Periodo 16 (reale)**
Alice	puntata	puntata	puntata
	30.000	30.600	30.600
	Slot assegnato	Slot assegnato	Slot assegnato
	600	600	600
	Controllo della rete	Controllo della rete	Controllo della rete
	2,5%	2,5%	2,5%
Mallory	puntata	puntata	puntata

	30.000	30.600	30,240
	Slot assegnato	Slot assegnato	Slot assegnato
	600	600	593 (-7) (-1,17%)
	Controllo della rete	Controllo della rete	Controllo della rete
	2,5%	2,5%	2,47% (-0,03%)
Altri partecipant	puntata	puntata	puntata
	1.140.000	1.162.800	1.162.800
	Slot assegnato	Slot assegnato	Slot assegnato
	22.800	22.800	22.807 (+7)
	Controllo della rete	Controllo della rete	Controllo della rete
	95%	95%	95,03% (+ 0,03%)

Come si può vedere, la potenza di calcolo di Mallory non è stata ridotta (l'algoritmo non è punitivo nel senso classico della rimozione di qualcosa) ma è cresciuto più lentamente di quello degli altri componenti della rete. Ciò ha causato, nel punto di ricalcolo del momento 16, un cambiamento nell'allocazione degli slot con Mallory per cui si è visto togliere lo 0,03% della potenza di calcolo rispetto al totale degli slot di rete. Invece, confrontandolo con il suo numero di slot precedente, la riduzione corrisponde alla perdita di 1,17% degli slot.

Secondo esempio: puntata squilibrata con commissione

Supponiamo che due indirizzi MAIN:

- Alice, alice_main_addr, node_a1, node_a2,…, puntata node_a10

  Puntata 30.000

- Mallory, mallory_main_addr, node_m1

  Puntata 30.000

Puntata totale: 1.200.000
Puntata minima: 3.000 Puntata massima: 6.000
Slot min: 60 Slot max: 120

Ad ogni blocco generato viene assegnato un premio di 1 token

Nel momento 14 Alice e Mallory hanno completato la registrazione dei loro indirizzi principali ed hanno assegnato i loro nodi.

Questi saranno attivi a partire dal periodo 15

Nel periodo 16 verrà calcolato il premio.

All'inizio del periodo 15 Alice avrà dieci nodi attivi (node_a1, ..., node_a10) ciascuno con 60 slot assegnati.

All'inizio del periodo 15, Mallory avrà un nodo node_m1 attivo con 600 slot assegnati.

Gli slot di Alice e Mallory sono gli stessi poiché hanno la stessa puntata attiva. Tuttavia, la distribuzione cambia. Mallory ha un solo nodo mentre Alice ne ha dieci. Mallory è penalizzato. I nodi di Alice sono attivi con il minimo, il nodo di Mallory è di 480 slot oltre il limite e per questo riceverà una penalità.

Supponendo una perfetta esecuzione del protocollo nel periodo 15, supponiamo che una media di 5 token/blocco sia l'ammontare della commissione raccolta, per maggior semplicità.

I dieci nodi di Alice hanno assegnati 60 slot, producendo così 60 blocchi, ed in ciascun blocco vi sarà un premio di 1 token. Essendo dieci i nodi (60 * 10 * 1) il premio sarà di 600 token. All'inizio del periodo 16, il saldo di Alice aumenterà da 30.000 a 30.600 token. A questi aggiungeremo i premi delle operazioni 600 * 5 = 3.000 token. Alice totalizzerà 33.600 token a seguito di operazioni nel periodo 15.

Passiamo ora a Mallory: il nodo di Mallory ha assegnati 600 slot, quindi produrrà 600 blocchi, e per i primi 120 blocchi verrà assegnato un premio di 1 token. Per i seguenti 480 il premio sarà 0. Per quanto riguarda i premi, usando la distribuzione omogenea indicata nelle condizioni preliminari, a Mallory verrà assegnato il premio solo nei primi 120 slot. Per questi riceverà 5 * 120 = 600 token. I restanti 2.400 vengono congelati per essere

restituiti in un periodo di 480 * 2 = 960 slot entro i limiti. Supponendo che la configurazione dei nodi non cambi dal periodo 15 al periodo 16, valutiamo cosa succede all'EEP del periodo 16 per il periodo 17.

All'inizio del periodo 16, il saldo di Mallory dovrebbe aumentare da 30.000 a 30.600 token a seguito della base monetaria e dovrebbero essere assegnati ulteriori 3.000 token a causa delle commissioni. Solo che Mallory ha attivato un numero di nodi insufficiente per soddisfare i requisiti della rete.

Nodo	**Premio raccolto**
node_m1	120 (coinbase) + 600 (a pagamento) *2400 sono congelati per 960 slot e non possono entrare nel calcolo del premio*
Totale	720

Passiamo ora a una valutazione della parte rimanente della rete.

Alice e Mallory hanno in totale (30.000 Alice e 30.000 Mallory) 60.000 token di puntata.

Sul resto della rete (1.200.000 - 60.000) = 1.140.000 token rimangono in gioco.

Supponendo che tutto il resto della rete funzioni in condizioni ideali e senza transazioni con gli altri nodi che compongono la rete, verranno assegnati:

Slot totali	Slot di Alice	Slot di Mallory	Slot assegnati al resto della rete
24000	-600	-600	= 22800

Verranno generati 22.800 blocchi e quindi un premio di 22.800 token per la parte della base monetaria. Per la parte di commissioni 114.000 token

Puntate precedenti di altri nodi	1.140.000 +
Altri nodi di base monetaria	22.800 +

Altri nodi di concessione	114.000	=		
	Totale altri nodi		***1.276.800***	***+***
Puntata precedente il periodo di Alice	30.000	+		
Base monetaria di Alice	600	+		
Commissione di Alice	3.000	=		
	Totale Alice		***33.600***	***+***
Puntata precedente il periodo di Mallory	30.000	+		
Base monetaria di Mallory	120	+		
Commissione di Mallory	600	=		
	Totale di Mallory		***30,720***	***=***
Puntata totale per nuovo periodo			**1.341.120**	

La tabella seguente confronta la percentuale di controllo della rete nel momento 15 con la rivalutazione nel periodo 16 (prevista) e 16 (reale)

	Periodo 15	**Periodo 16 (previsto)**	**Periodo 16 (reale)**
Alice	puntata	puntata	puntata
	30.000	33.600	33.600
	Slot assegnato	Slot assegnato	Slot assegnato
	600	600	601 (+1)
	Controllo della rete	Controllo della rete	Controllo della rete
	2,5%	2,5%	2,51% (+0,01)
Mallory	puntata	puntata	puntata

	30.000	33.600	30,720
	Slot assegnato	Slot assegnato	Slot assegnato
	600	600	550 (-50) (-8,33%)
	Controllo della rete	Controllo della rete	Controllo della rete
	2,5%	2,5%	2,29% (-0,21%)
Altri partecipanti	puntata	puntata	puntata
	1.140.000	1.276.800	1.276.800
	Slot assegnato	Slot assegnato	Slot assegnato
	22.800	22.800	22.849 (+49)
	Controllo della rete	Controllo della rete	Controllo della rete
	95%	95%	95,20% (+ 0,20%)

Come si può vedere, il potere computazionale di Mallory non è stato ridotto (l'algoritmo non è punitivo nel senso classico della rimozione di qualcosa) ma è cresciuto più lentamente di quello degli altri componenti della rete. Ciò ha causato, nel punto di ricalcolo del periodo 16, un cambiamento nell'allocazione degli slot con Mallory che si è visto togliere lo 0,21% della potenza di calcolo sulla rete. Confrontandolo con il suo numero di slot precedente, la riduzione corrisponde alla perdita dell'8,33% dello slot. L'algoritmo, in presenza delle commissioni (quindi di una rete caricata), diventa fortemente punitivo. Questo perché quanto maggiore è il carico della rete, tanto più le violazioni diventano critiche.

Terzo esempio: Recupero di una puntata congelata

In questa sezione verrà presentato il meccanismo di restituzione per le puntate che sono state congelate a partire dalla condizione simulata nella puntata squilibrata con

commissione. In questo modo viene premiato il rientro nei parametri operativi ottimali della rete.

Mallory riconfigura il pool per adattarlo ai parametri di mining ottimali

A partire dalla condizione sopra menzionata (secondo esempio), prima dell'EEP del periodo 16 esso aggiunge 4 nodi. Con l'aggiunta di 4 nodi i suoi 550 slot sono divisi in 550/5 = 110 <120. Come sempre, la modifica avrà effetto solo dal prossimo periodo.

Con questa modifica, esso lascia completamente la zona di penalità.

Come condizioni iniziali si considerino quelle evidenziate in grigio dell'ultima colonna.

Si simuli ora l'esecuzione nel caso in cui Mallory corregga il suo comportamento con l'aggiunta di 4 nodi.

In questo caso si otterrà una premio per i suoi 550 slot:

coinbase = 550 * 1 = 550

collected_fee = 550 * 5 = 2750

Dal momento precedente, Mallory ha ereditato una commissione congelata di 2400 token ed un numero di slot di penalità pari a 960 (il doppio degli slot per i quali esso è rimasto al di fuori dei parametri).

La formula per calcolare la commissione recuperata per uno slot è la seguente:

recovered_feeE17S1 = frozen_feeE17S1/penalty_slotsE17S1

A questo punto vengono aggiornati i valori di frozen_fee e di penalty_slots:

frozen_feeE17S2 = frozen_feeE17S1 - recovered_feeE17S1

penalty_slotsE17S2 = penalty_slotsE17S1 - 1

Per comodità di nota, il periodo del valore assoluto è indicato nel pedice, in questo caso 17, ma nello slot 2 è indicato lo slot assegnato a Mallory dopo uno. I due slot potrebbero non essere consecutivi. In questo caso, dopo la "s", viene mostrato il numero progressivo dello slot assegnato a Mallory.

La tabella seguente mostra l'evoluzione dei valori al passaggio degli slot.

Slot assegnato	frozen_fee	penalty_slots	recovered_fee
E17S1	2400	960	2,5
E17S2	2397,5	959	2,5
E17S3	2395	958	2,5
...			
E17S109	2130	852	2,5
E17S110	2127,5	851	2,5

Alla fine del periodo, Mallory avrà recuperato 275 token per i 2400 congelati. Ciò gli consente di colmare parzialmente il divario con gli altri nodi della rete e costituisce un incentivo a favorire il comportamento virtuoso di aver aggiunto nodi.

Nelle commissioni congelate non vi sono basi monetarie che non siano state generate. È sempre ed in ogni caso più conveniente per un nodo operare in conformità con le regole piuttosto che utilizzare questa procedura di ripristino.

Secondo una forma preferita di realizzazione, inoltre, un accesso ad una risorsa condivisa è calcolato in modo autonomo da tutti i nodi e.g. nodi OVERFLOW, della rete. A tale riguardo, nelle blockchain del tipo proof of stake, è di importanza non trascurabile garantire una opportuna redistribuzione dei compiti di costruzione (denominato "mining") dei blocchi. Mentre risulta semplice determinare in modo centralizzato l'ordine di assegnazione dei blocchi da minare tra i nodi della blockchain, una determinazione delle assegnazioni in modo completamente distribuito risulta alquanto complesso.

I sistemi di determinazione dell'ordine di accesso ad un mezzo trasmissivo a divisione di tempo sono particolarmente noti come TDMA dall'acronimo anglosassone "Time division multiple access". Questi sistemi sono stati congeniati nell'ambito delle telecomunicazioni per evitare collisioni tra messaggi generati da più nodi, che portano ad un rapido decremento della capacità della rete, detto "throughput".

L'approccio TDMA prevede di suddividere il tempo in macro intervalli denominati epoche, in cui ciascuna epoca è suddivisa in cosiddetti time slot. Il numero dei time slot

attribuiti a ciascun nodo che condivide una risorsa condivisa è predeterminato secondo una predeterminata logica di distribuzione.

Con riferimento alla figura 11 ciascun nodo N1, N2, N4, etc., che accede alla risorsa condivisa ha un proprio identificativo univoco ID1, ID2, ID4, etc.. generalmente numerico.

Gli identificativi possono essere ordinati secondo una lista, così come mostrata in figura 11.

A ciascun identificativo è associato un numero di time slot a cui il nodo ha diritto di accesso in una predeterminata epoca. Un'epoca è un raggruppamento ordinato nel tempo, di time slot.

A ciascun nodo è anche associato un intervallo numerico di ampiezza proporzionale al numero dei time slot associati allo stesso nodo.

Pertanto, se N1 ha diritto di accesso a due time slot il relativo intervallo numerico è ad esempio 20, mentre se N2 ha diritto di accesso ad un solo time slot il relativo intervallo numerico è di 10.

I vari intervalli numerici sono contigui e non si sovrappongono.

Ad esempio, per N1 l'intervallo va da 0 a 19. Per N2, l'intervallo va da 19 a 29. Per N4 l'intervallo va da 29 a 59.

Gli altri nodi N3 ed N5, evidentemente, non hanno diritto ad accedere alla risorsa condivisa nell'epoca presa in esame.

L'intervallo numerico complessivo è dato dalla somma SUM.

Dal momento che ciascun nodo conosce

- l'ID degli altri nodi,
- il numero dei time slot cui ciascun nodo ha diritto di accesso,
- il seme,

allora può calcolare autonomamente l'ordine di accesso alla risorsa condivisa secondo i passi eseguiti ciclicamente per un numero di cicli pari al numero di time slot dell'epoca:

- i-mo calcolo dell'Hash del seme,
- Calcolo del resto della divisione di detto Hash per la somma SUM delle ampiezze di detti intervalli numerici,
- Determinazione dell'intervallo in cui cade detto resto,
- Attribuzione dell'i-mo slot al nodo corrispondente a detto intervallo.

Ad esempio se l'Hash del seme porta ad un valore numerico di 600, allora, dal momento che SUM è pari a 60, la divisione 610/60 porta come resto 10. Il valore 10 cade nel primo intervallo numerico R1 che appartiene ad N1. Pertanto il primo time slot è assegnato ad N1.

Viene dunque calcolato l'Hash dell'Hash precedentemente calcolato. Questa volta, il resto della divisione porta ad un valore pari a 32 che cade nell'intervallo R4 che appartiene ad N4. Pertanto il secondo time slot appartiene ad N4.

Dal momento che R4 possiede un intervallo maggiore rispetto agli altri, la probabilità di vedersi assegnare un time slot è proporzionale all'ampiezza del proprio intervallo e pertanto, si garantisce la corretta redistribuzione di time slot, ma con un ordine di assegnazione massimamente dipendente dal seme.

La procedura prosegue fintanto che tutti i time slot sono assegnati.

Per quanto concerne la divisione, si deve tener conto che un Hash può avere una rappresentazione numerica qualunque, ad esempio mZOhnZ2ZSUva+Cq16HEy0+1I87DHFhVUKI28QcswrfQ= e può essere convertita in notazione esadecimale diventando 9993a19d9d99494bdaf82ab5e87132d3ed48f3b0c7161554288dbc41cb30adf4 od in notazione decimale diventando 391875681579805356411831496321815245512789534030969319853065217075543948136635096802576932411078811361721 3.

Pertanto, la suddetta divisione può essere realizzata eseguendo eventuali conversioni di notazione del divisore o dividendo affinché siano compatibili.

Ciò si applica anche a qualunque identificativo univoco.

La figura 14 mostra un diagramma di flusso esplicativo del metodo oggetto della presente invenzione, il quale comprende i seguenti passi preliminari in successione:

- (i) condivisione tra tutti i nodi di tutti gli identificativi univoci di almeno detto predeterminato numero di nodi (N1, N2, N4) avente diritto di accesso alla risorsa condivisa,
- (ii) condivisione di detto rispettivo predeterminato numero di time slot assegnato a ciascun nodo,
- (iii) condivisione di un seme;

il metodo comprendendo i seguenti passi in successione

- (iv) ordinamento di detti identificativi,
- (v) associazione a ciascun nodo di un intervallo numerico (R1, R2, R4) di ampiezza proporzionale a detto rispettivo predeterminato numero di time slot associati,
- (vi) allineamento di detti intervalli numerici in modo da risultare contigui e senza sovrapposizione secondo un ordine definito da detto ordinamento di identificativi,
- (vii) calcolo di una somma (SUM) di detti intervalli numerici,
- (viii) esecuzione ciclica dei seguenti passi per un numero di cicli pari al numero di time slot dell'epoca:

    a. i-mo calcolo dell'i-mo Hash del seme,
    b. Calcolo del resto di una divisione di detto i-mo Hash per detta somma (SUM),
    c. Determinazione dell'intervallo numerico (R1, R2, R4) in cui cade detto resto,
    d. Attribuzione dell'i-mo slot al nodo corrispondente a detto intervallo numerico in cui cade detto resto.

La figura 13 mostra uno schema di attribuzione casuale dei time slot secondo la presente invenzione, descritto ora ipotizzando che la risorsa condivisa è un mezzo trasmissivo.

Si ipotizza che esistono un predeterminato numero di stazioni trasmittenti N1, N2, N4, indicate come "Nodes" ed un altro numero A, B, C di generatori di flusso, indicati come

"Holders". Deve essere chiaro che è solo un caso che le stazioni trasmittenti ed i generatori di flusso siano in uguale numero.

Gli Holders, ad esempio possono essere server di accesso ad una porzione di una rete telematica, per esempio internet.

Il generatore di flusso A ha diritto di utilizzare tre time slot, il generatore di flusso B ha diritto a cinque time slot, mentre il generatore di flusso C ha diritto a sette time slot.

Evidentemente, ciascun generatore di flusso può inoltrare richiesta di accesso al mezzo trasmissivo ad una stazione trasmittente.

A desidera richiedere al nodo N1 di trasmettere i propri dati durante i time slot assegnati.

B distribuisce due incarichi ad N1, tre in carichi ad N2 e due and N3.

Il calcolo dei time slot da accedere può essere eseguito dai generatori di flusso oppure dalle stazioni trasmittenti. Inoltre, gli identificatici ID possono appartenere ai generatori di flusso oppure ai nodi trasmissivi.

Pertanto, i time slot dell'epoca (Epoch) di figura 3 sono assegnati mediante la presente invenzione ad N2, N1, N3,N2, N3, etc..

La figura 13 può essere interpretata, anche come una situazione in cui in una blockchain del tipo proof of stake, i generatori di flusso, indicati in figura come "Holders" sono server che ricevono/collezionano richieste di transazioni, mentre i nodi N1 - N4 corrispondono ai minatori.

In entrambi i casi, gli Holders possono scegliere il nodo a cui affidare le proprie transazioni (od il proprio flusso di dati), in relazione alla affidabilità dello stesso nodo. Per esempio, tenendo traccia della efficienza nell'esecuzione di compiti assegnati in epoche precedenti.

La presente invenzione può essere vantaggiosamente realizzata tramite un programma per computer che comprende mezzi di codifica per la realizzazione di uno o più passi del metodo, quando questo programma è eseguito su di un computer. Pertanto si intende che l'ambito di protezione si estende a detto programma per computer ed inoltre a mezzi

leggibili da computer che comprendono un messaggio registrato, detti mezzi leggibili da computer comprendendo mezzi di codifica di programma per la realizzazione di uno o più passi del metodo, quando detto programma è eseguito su di un computer.

RIVENDICAZIONI

1. Metodo implementato da computer per raggiungere un consenso distribuito basato su una proof of stake per una rete blockchain, permettendo così di raggiungere un livello più alto di distribuzione della rete e di non avere troppe puntate concentrate in troppi nodi, detta rete blockchain comprendendo uno o più indirizzi MAIN, per ciascun indirizzo MAIN essendo associato a sua volta uno o più indirizzi/nodi OVERFLOW, l'ultimo essendo in grado di fungere da potenziali nodi di mining solo se una quantità minima prestabilita di puntate è assegnata a detto uno o più indirizzi/nodi OVERFLOW, comprendente i seguenti processi:

-un processo di ridistribuzione configurato per calcolare la puntata minima necessaria per un nodo OVERFLOW da estrarre, essendo minimum_stake_value = total_amount_at stake/predetermined_target_number_of_nodes, per calcolare la puntata massima che consente un mining efficiente, essendo il valore di puntata massimo = valore di puntata minimo * 2, per ciascun indirizzo MAIN per distribuire la puntata tra i nodi OVERFLOW, scorrendo attraverso i nodi OVERFLOW di un dato indirizzo MAIN e per assegnare a ciascun nodo OVERFLOW il minimum_stake_value, fino a quando la puntata rimanente non è sufficiente per attivare un miner o per fare in modo che a ciascun nodo di overflow sia stata assegnata la sua puntata;

-un processo di penalità configurato per congelare una quantità di ricompensa corrispondente al lavoro computazionale per aver minato un numero di slot su un limite prestabilito Slot_MAX in un $periodo_{i+1}$ e per recuperarlo nei periodi seguenti a partire dal $periodo_{i+2}$, un po' alla volta, in base ad un numero di parti proporzionale al numero di slot minati su detto limite prestabilito Slot_MAX, essendo una parte assegnata a ciascuno slot successivo per il quale il miner è delegabile al mining, quindi al fine di incentivare naturalmente la distribuzione della rete blockchain.

2. Metodo implementato da computer secondo la rivendicazione 1 in cui il processo di ridistribuzione può venire ulteriormente configurato in modo tale per cui, se rimane una

qualsiasi puntata, essa viene distribuita tra i nodi OVERFLOW fino a quando il maggior numero possibile di nodi ha una puntata pari al valore massimo della puntata, e ciascuna ulteriore puntata viene assegnata al primo nodo OVERFLOW secondo un ordine prestabilito.

3. Metodo implementato da computer secondo una delle rivendicazioni precedenti in cui il processo di ridistribuzione comprende in particolare le seguenti fasi:

52: avere una rete blockchain comprendente uno o più indirizzi MAIN, per ciascun indirizzo MAIN essendo associati a loro volta uno o più indirizzi/nodi di OVERFLOW che agiscono come potenziali nodi di mining solo se una quantità minima prestabilita di puntate è assegnata a detto uno o più indirizzi/nodi di OVERFLOW;

55: ottenere come valore TOTAL_AT_STAKE di input per ciascun indirizzo MAIN della rete blockchain corrispondente alla quantità totale di puntate su ciascun indirizzo MAIN

60: calcolo di S_min_VAL = TOTAL AT STAKE/TARGET NUMBER OF_NODES essendo TARGET NUMBER OF_NODES = numero totale di nodi di mining OVERFLOW nella rete della chain,

Calcolo di S_MAX_VAL = S_min_VAL * 2

Supponendo un valore di contatore ITERATION = 0

65: se il valore ITERATION è <2, aggiornare il valore del contatore ITERATION = ITERATION + 1

70: per ciascun indirizzo MAIN che ottiene come input il valore MAIN_ADDRESS.assigned_stake e se questo valore è> 0, il che significa che l'indirizzo MAIN in esame contiene una quantità di puntata, assegnando questo valore al parametro M_A_S,

e se il valore M_A_S è> = S_MAX_VAL ripetere l'operazione 70 fino al successivo indirizzo MAIN,

75: se il valore M_A_S è <S_MAX_VAL ottenere come input il valore OVERFLOW_ADDRESS.assigned_stake per ciascun indirizzo OVERFLOW associato al MAIN in esame, e

se il valore OVERFLOW_ADDRESS.assigned_stake è = 0 ripetere l'operazione 75 fino al successivo indirizzo OVERFLOW associato al MAIN in esame e

se il valore OVERFLOW_ADDRESS.assigned_stake è ≠ 0, assegnare il valore OVERFLOW_ADDRESS.assigned_stake al parametro O_A_S, calcolando MAIN_ADDRESS.assigned_stake = M_A_S - S_min_VAL, e calcolando OVERFLOW_ADDRESS.assigner_stake = O_A_S + S_min_VAL, e ripetendo il punto 75 al successivo indirizzo OVERFLOW associate MAIN in esame.

4. Metodo implementato da computer secondo una delle precedenti rivendicazioni 1, 2 in cui il processo di penalità comprende in particolare le seguenti fasi:

80: calcolare il totale frozen_fee $_{i+1}$ come importo del premio$_{i+2}$ non erogato ai miners in un periodo specifico $_{i+1}$ per aver minato un numero di slot oltre il limite Slot_MAX

85: calcolare i penalty_slots$_{i+1}$ preferibilmente per due volte (o secondo un'altra legge proporzionale) il numero di slot minati oltre il limite, cioè il numero di slot su Slot_MAX

90: calcolare il valore recovered_fee $_{j, i+1}$ come frozen_fee$_{i+1}$/penalty_slots$_{i+1,}$ ovvero la commisssione da recuperare per ogni slot

95: aggiungere il valore recovered_fee$_{j, i+1}$ al premio del MAIN_address per il mining del blocco $_{i}$ nel periodo$_{i+1}$ come premio$_{j, i+1}$ = un coinbase + collect_fee $_{j, i+1}$ + recovered_fee $_{j, i+1}$

100: calcolare il nuovo valore del parametro frozen_fee $_{i+1}$ come frozen_fee $_{i+1}$ - recovered_fee $_{j, i+1}$ ed il nuovo valore di penalty_slot $_{i+1}$ come penalità_slot $_{i+1}$ - 1 e ripetere l'operazione 95 per il seguente blocco $_{j+1}$ nel periodo $_{i+1}$ mentre frozen_fee $_{i+1}$ > 0 o mentre penalty_slot $_{i+2}$ > 0, per cui quando l'ultimo blocco$_{j}$ viene minato alla fine di ogni periodo$_{i}$, penalty_slots$_{i}$ e frozen_fee$_{i}$ vengono trasferiti dal primo nodo

OVERFLOW al suo indirizzo MAIN e quando il primo blocco $_j$ viene minato all'inizio di ogni periodo$_i$, penalty_slots$_i$ e frozen_fee$_i$ vengono trasferiti dall'indirizzo MAIN al suo primo nodo OVERFLOW.

5. Metodo implementato da computer secondo una delle rivendicazioni precedenti comprendente inoltre un processo di inizializzazione configurato per elencare tutti gli indirizzi MAIN della rete blockchain, per ciascun elenco di indirizzi MAIN i suoi indirizzi/nodi OVERFLOW assegnati e quindi per ciascun indirizzo OVERFLOW/il nodo trasferisce la puntata, se presente, all'indirizzo MAIN assegnato.

6. Metodo implementato da computer secondo la rivendicazione 5 in cui il processo di inizializzazione comprende in particolare le seguenti operazioni:

20: avere una rete blockchain comprendente uno o più indirizzi MAIN, per ogni indirizzo MAIN essendo associati a loro volta uno o più indirizzi/nodi OVERFLOW che agiscono come potenziali nodi di mining solo se una quantità minima predeterminata di puntate è assegnata a detto uno o più indirizzi OVERFLOW/nodo;

25: tramite la rete blockchain, ricevere una transazione includendo come indirizzi di input dal registro di indirizzi che comprende gli indirizzi MAIN ed i loro corrispondenti indirizzi OVERFLOW;

30: associare a ciascun indirizzo MAIN un numero progressivo univoco;

35: per ciascun indirizzo MAIN verificare se esso contiene una quantità di puntate e se questo valore è> 0 assegnare questo valore al parametro M_A_S, e se questo valore è <= 0 eseguire l'operazione 35 fino al successivo indirizzo MAIN membro della rete blockchain e generare una transazione incluso l'output che fornisce il valore M_A_S associato al corrispondente indirizzo MAIN,

40: associare a ciascun nodo OVERFLOW un numero progressivo univoco

45: per ciascun nodo OVERFLOW assegnare a ciascun indirizzo MAIN corrispondente verificando se contiene una quantità di puntate definita OVERFLOW_ADDRESS.assigned_stake e se questo valore è> 0 assegnare questo

valore al parametro O_A_S, calcolando il valore MAIN_ADDRESS.assigned_stake come M_A_S + O_A_S e assegnando O_A_S = 0 e generare una transazione che include l'output che fornisce il valore aggiornato del valore M_A_S associato al corrispondente indirizzo MAIN, quindi eseguire l'operazione 45 fino al successivo indirizzo OVERFLOW assegnato al corrispondente indirizzo MAIN considerato nell'operazione 35,

e se OVERFLOW_ADDRESS.assigned_stake è <= 0 eseguire l'operazione 45 fino al successivo indirizzo OVERFLOW assegnato al corrispondente indirizzo MAIN considerato fino all'operazione 35

50: generare una transazione che include l'output del valore aggiornato del valore TOTAL_AT_STAKE per ciascun indirizzo MAIN della rete blockchain corrispondente alla quantità totale di puntate su ciascun indirizzo MAIN dopo aver eseguito le operazioni precedenti.

7. Metodo implementato da computer secondo una delle rivendicazioni precedenti comprendente inoltre un processo di registrazione configurato per registrare uno o più indirizzi MAIN, per registrare uno o più indirizzi/nodi OVERFLOW e per associare uno o più indirizzi/nodi OVERFLOW a detto uno o più indirizzi MAIN e registrare detta associazione.

8. Metodo implementato da computer secondo una delle rivendicazioni precedenti che comprende inoltre un processo di manutenzione, al fine di mantenere la rete blockchain altamente distribuita anche in caso di guasto di uno o più nodi di mining, detto processo di manutenzione comprendente a sua volta le seguenti operazioni, dato un indirizzo MAIN ed almeno due o più indirizzi OVERFLOW e un nodo che si desidera manutenere:

110: inviare una transazione che include un primo input che fornisce un indirizzo OVERFLOW sostitutivo usando il suo codice privato, il che significa che tale indirizzo OVERFLOW sostitutivo è un nodo che sostituisce il nodo che si desidera manutenere

120: assegnare detto indirizzo OVERFLOW sostitutivo al corrispondente indirizzo MAIN utilizzando il codice privato dell'indirizzo MAIN

130 annullare la registrazione del nodo che si desidera manutenere utilizzando il suo codice privato

9. Metodo secondo una qualsiasi delle precedenti rivendicazioni distribuito per determinare un ordine di accesso ad una risorsa condivisa da parte di un predeterminato numero di detti nodi OVERFLOW (N1, N2, N4) aventi diritto di accesso alla risorsa condivisa in una predeterminata epoca (epoch) per un rispettivo predeterminato numero di time slot, in cui un'epoca è formata da una pluralità di time slot, ed in cui ciascun nodo (N1, N2, N4, etc.), ha un proprio identificativo univoco (ID1, ID2, ID4), il metodo comprendendo inoltre i seguenti ulteriori passi preliminari in successione:

- (i) condivisione tra tutti i nodi di tutti gli identificativi univoci di almeno detto predeterminato numero di nodi (N1, N2, N4) avente diritto di accesso alla risorsa condivisa,
- (ii) condivisione di detto rispettivo predeterminato numero di time slot assegnato a ciascun nodo,
- (iii) condivisione di un seme;

il metodo comprendendo i seguenti passi in successione

- (iv) ordinamento di detti identificativi,
- (v) associazione a ciascun nodo di un intervallo numerico (R1, R2, R4) di ampiezza proporzionale a detto rispettivo predeterminato numero di time slot associati,
- (vi) allineamento di detti intervalli numerici in modo da risultare contigui e senza sovrapposizione secondo un ordine definito da detto ordinamento di identificativi,
- (vii) calcolo di una somma (SUM) di detti intervalli numerici,
- (viii) esecuzione ciclica dei seguenti passi per un numero di cicli pari al numero di time slot dell'epoca:

a. i-mo calcolo dell'i-mo Hash del seme,
b. Calcolo del resto di una divisione di detto i-mo Hash per detta somma (SUM),

c. Determinazione dell'intervallo numerico (R1, R2, R4) in cui cade detto resto,
d. Attribuzione dell'i-mo slot al nodo corrispondente a detto intervallo numerico in cui cade detto resto.

10. Metodo secondo la rivendicazione 9, in cui detta risorsa condivisa è un mezzo trasmissivo.

11. Metodo secondo la rivendicazione 10, in cui detta risorsa condivisa consiste nel compito di generare un blocco da condividere di una blockchain.

12. Metodo secondo la rivendicazione 11, in cui detto seme è dato dalla concatenazione di due o più hash di blocchi precedentemente costruiti ed opportunamente selezionati.

13. Metodo secondo la rivendicazione 12, in cui detti blocchi sono selezionati secondo i seguenti passi:

- ripartire i blocchi generati in un'epoca precedente in tre o più gruppi
- estrazione del primo blocco di ciascun gruppo oppure il primo, secondo e terzo hash del secondo gruppo oppure l'ultimo, penultimo e terzultimo blocco del primo gruppo di blocchi;
- concatenamento degli Hash relativi ai blocchi estratti al passo precedente.

14. Nodo configurato per implementare un metodo per raggiungere un consenso distribuito basato su una proof of stake per una rete blockchain, permettendo così di raggiungere un livello più alto di distribuzione della rete e di non avere troppe puntate concentrate in troppi nodi, detto nodo comprendendo un dispositivo di interfaccia, un processore accoppiato a detto dispositivo di interfaccia, una memoria accoppiata al processore, la memoria avendo immagazzinato su di esso istruzioni eseguibili da computer che, quando eseguite, configurano il processore per eseguire le corrispondenti operazioni del metodo implementato da computer di una qualsiasi delle rivendicazioni 1 a 13.

15. Programma di computer che comprende mezzi di codifica di programma atti a realizzare tutti passi o processi di una qualunque delle rivendicazioni da 1 a 13, quando detto programma è fatto girare su di un computer.

16. Supporto di memorizzazione leggibile da computer comprendenti un programma registrato, detti mezzi leggibili da computer comprendendo mezzi di codifica di programma atti a realizzare tutti passi o processi di una qualunque delle rivendicazioni da 1 a 13, quando detto programma è fatto girare su di un computer.

17. Rete trasmissiva comprendente una pluralità di stazioni trasmittenti atte a condividere un mezzo trasmissivo, in cui ciascuna stazione trasmittente è configurata per eseguire tutti i passi e processi del metodo secondo le rivendicazioni 1 e 9 per determinare i time slot in cui ha diritto di accesso al mezzo trasmissivo.

18. Rete di computer per eseguire una transazione basata su blockchain da un primo utente a un secondo utente in una rete di trasferimento elettronico di nodi connessi, in cui i nodi includono almeno un primo nodo, un secondo nodo, secondo la rivendicazione 14, in cui ciascun nodo condivide una stessa blockchain, ed in cui ciascun nodo memorizza un supporto di memorizzazione leggibile da computer secondo la rivendicazione 16.

19. Rete secondo la rivendicazione 18, in cui ciascun nodo è atto a generare un blocco di dati da condividere, senza contesa, in cui ciascun nodo minatore è configurato per determinare i time slot assegnati per la generazione di blocchi della blockchain secondo il metodo della rivendicazione 9.

Riassunto

La presente invenzione riguarda un metodo implementato da computer ed un nodo che implementa il metodo per raggiungere un consenso distribuito basato su proof of stake, per un registro distribuito, come la tecnologia blockchain.

5

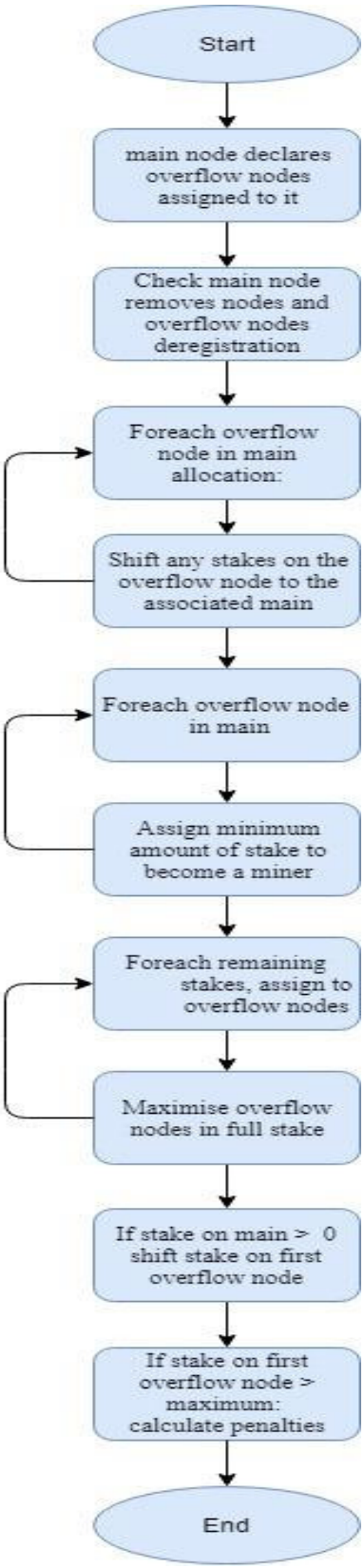
Start
main node declares overflow nodes assigned to it
Check main node removes nodes and overflow nodes deregistration
Foreach overflow node in main allocation:
Shift any stakes on the overflow node to the associated main
Foreach overflow node in main
Assign minimum amount of stake to become a miner
Foreach remaining stakes, assign to overflow nodes
Maximise overflow nodes in full stake
If stake on main > 0 shift stake on first overflow node
If stake on first overflow node > maximum: calculate penalties
End

Fig. 1

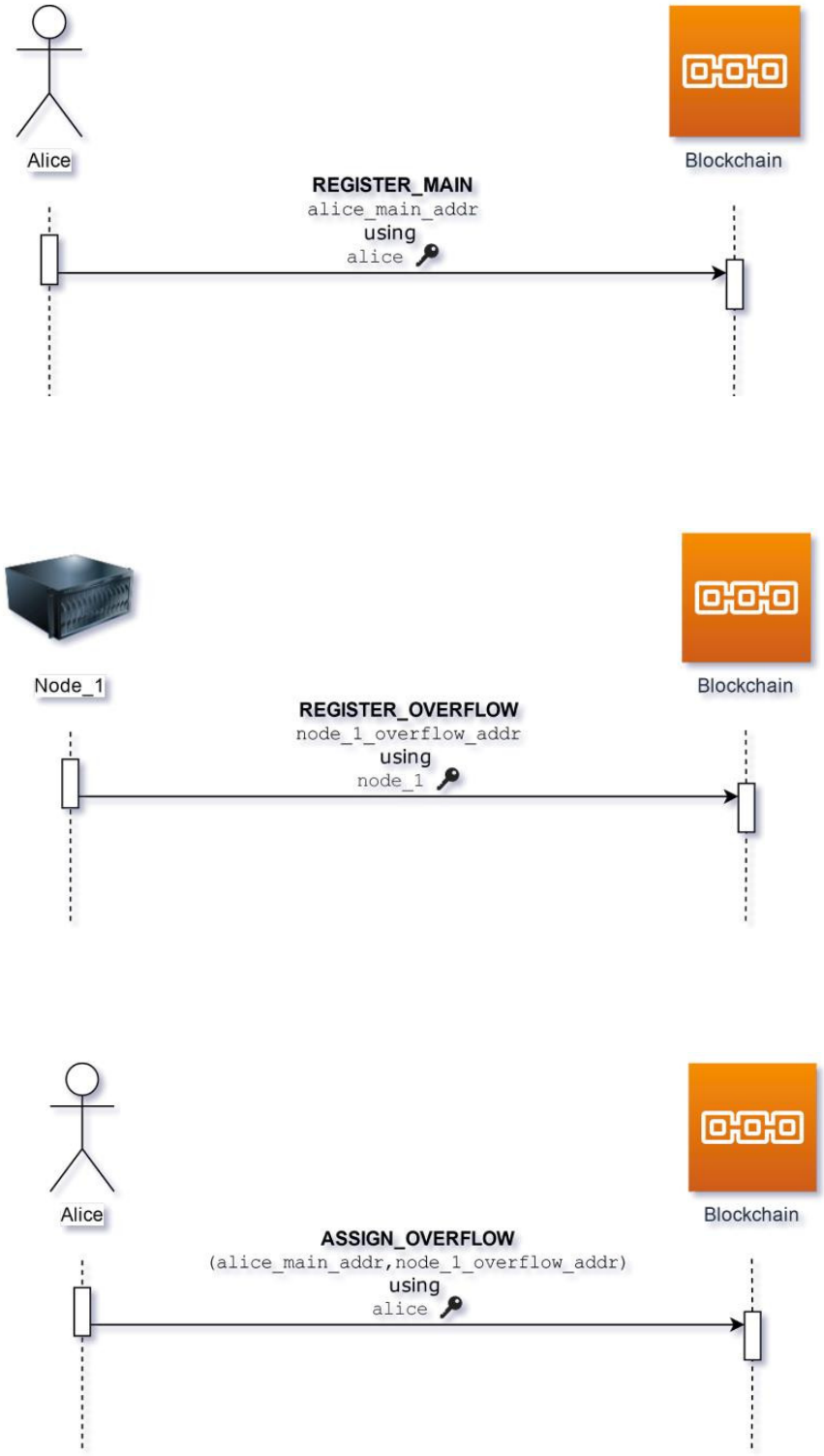
Alice
Blockchain
REGISTER_MAIN
alice_main_addr
using
alice
Node_1
Blockchain
REGISTER_OVERFLOW
node_1_overflow_addr
using
node_1
Alice
Blockchain
ASSIGN_OVERFLOW
(alice_main_addr,node_1_overflow_addr)
using
alice

Fig. 2

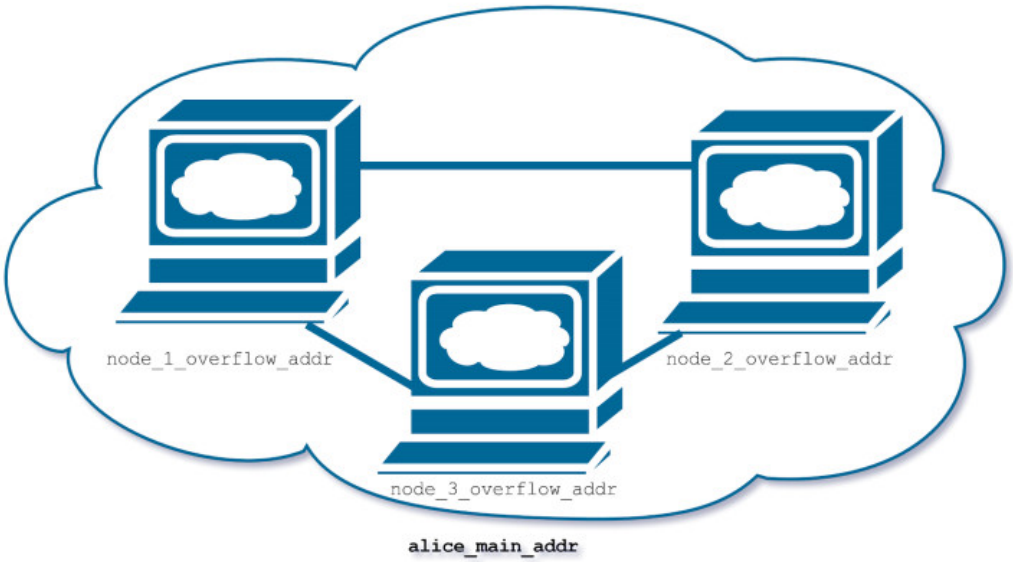
node_1_overflow_addr
node_2_overflow_addr
node_3_overflow_addr
alice_main_addr

Fig. 3

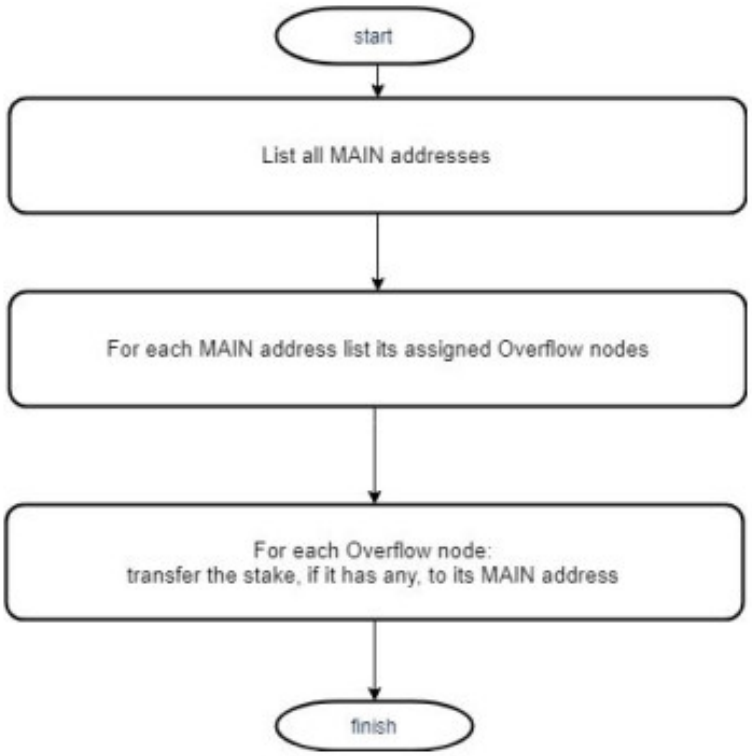
start
List all MAIN addresses
For each MAIN address list its assigned Overflow nodes
For each Overflow node:
transfer the stake, if it has any, to its MAIN address
finish

Fig. 4

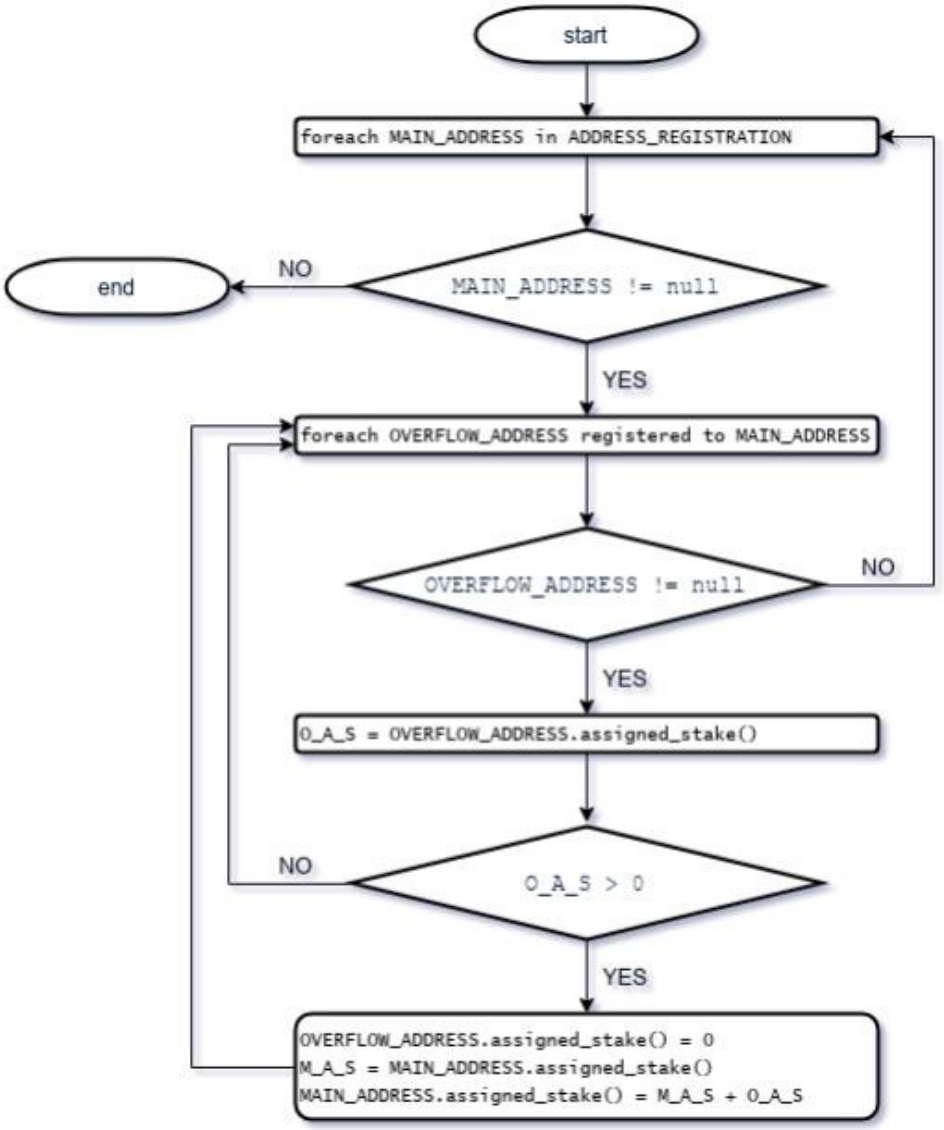

Fig. 5

INITIAL STATE								
	not miner	full pay	penalty area					
MAIN	S_min_VAL	S_min_VAL	S_min_VAL	S_min_VAL	S_min_VAL	S_min_VAL	S_min_VAL	RMD
OVERFLOW	0							
OVERFLOW	0							
OVERFLOW	0							

Fig. 6

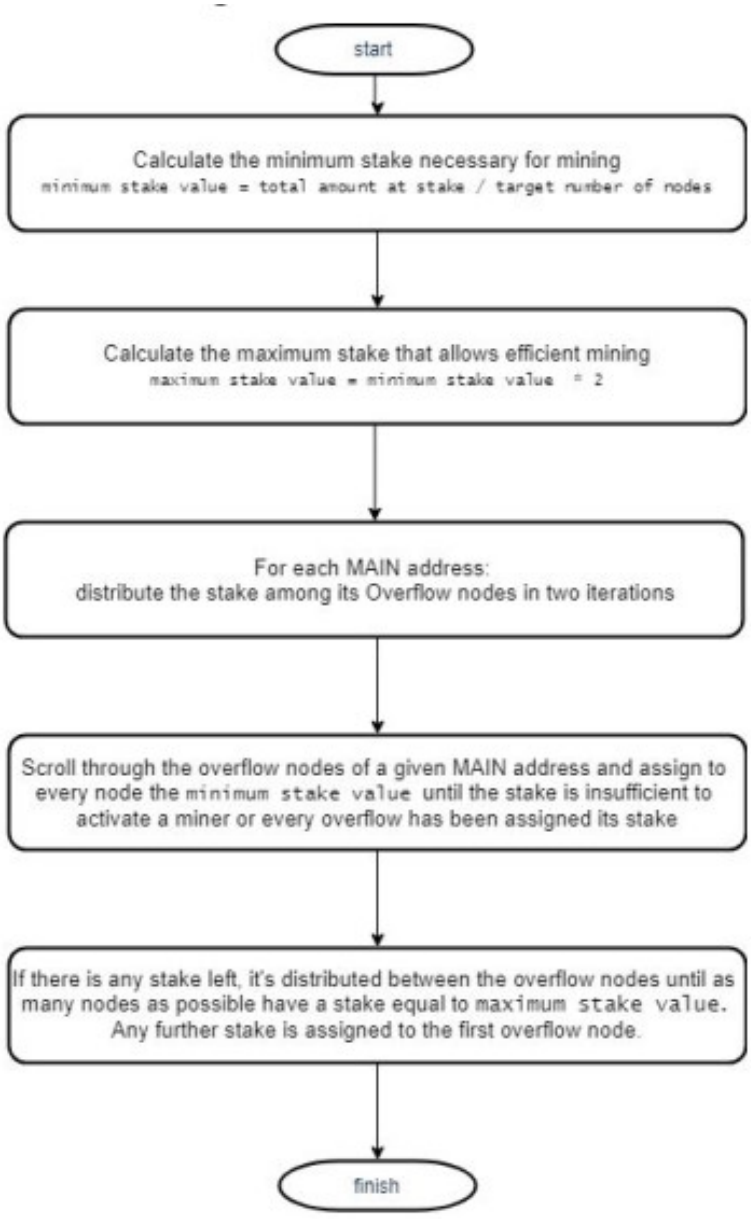
start
Calculate the minimum stake necessary for mining
minimum stake value = total amount at stake / target number of nodes
Calculate the maximum stake that allows efficient mining
maximum stake value = minimum stake value * 2
For each MAIN address:
distribute the stake among its Overflow nodes in two iterations
Scroll through the overflow nodes of a given MAIN address and assign to every node the minimum stake value until the stake is insufficient to activate a miner or every overflow has been assigned its stake
If there is any stake left, it's distributed between the overflow nodes until as many nodes as possible have a stake equal to maximum stake value. Any further stake is assigned to the first overflow node.
finish

Fig. 7

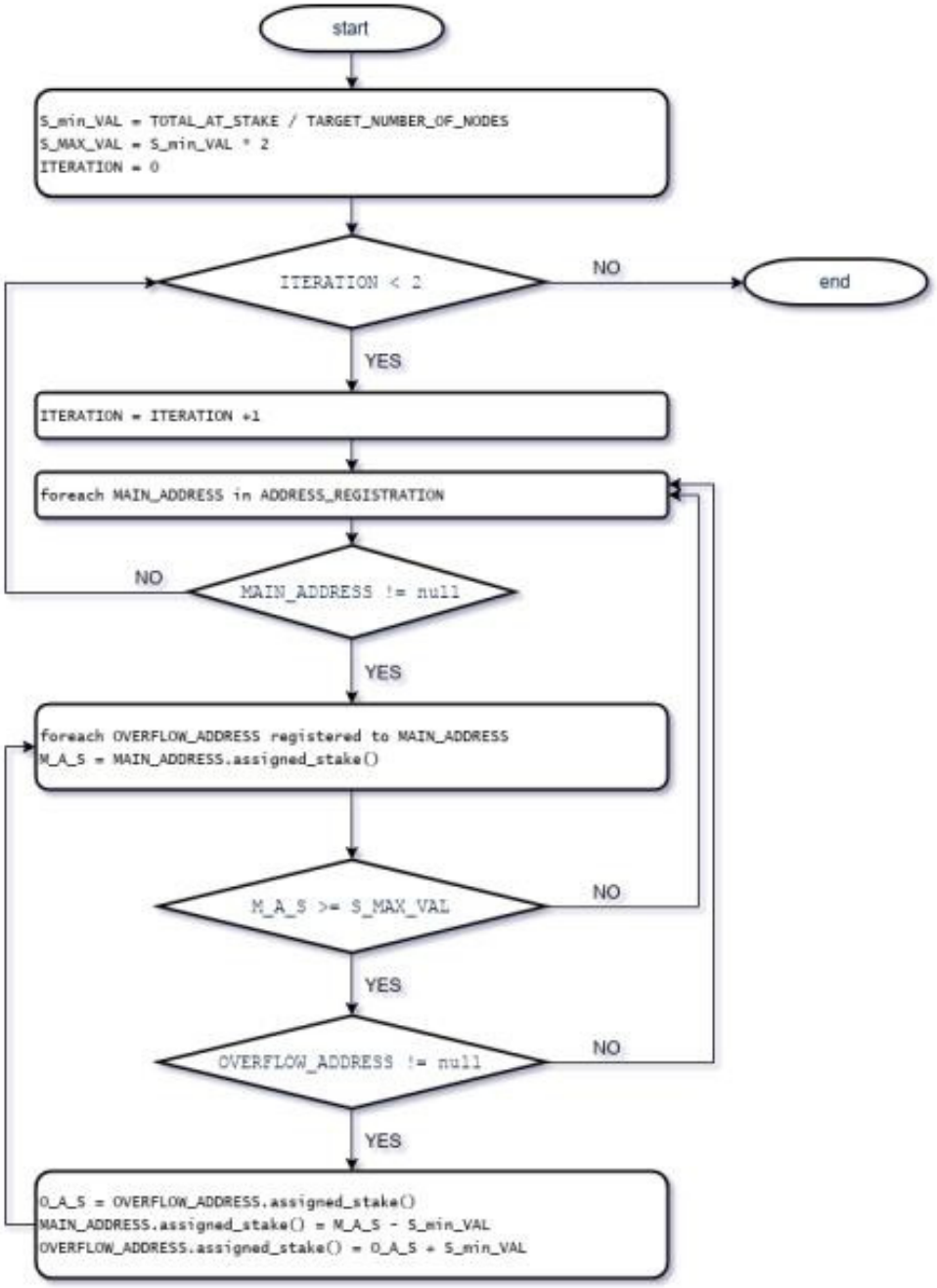
start
S_min_VAL = TOTAL_AT_STAKE / TARGET_NUMBER_OF_NODES
S_MAX_VAL = S_min_VAL * 2
ITERATION = 0
ITERATION < 2
NO
end
YES
ITERATION = ITERATION +1
foreach MAIN_ADDRESS in ADDRESS_REGISTRATION
NO
MAIN_ADDRESS != null
YES
foreach OVERFLOW_ADDRESS registered to MAIN_ADDRESS
M_A_S = MAIN_ADDRESS.assigned_stake()
M_A_S >= S_MAX_VAL
NO
YES
OVERFLOW_ADDRESS != null
NO
YES
O_A_S = OVERFLOW_ADDRESS.assigned_stake()
MAIN_ADDRESS.assigned_stake() = M_A_S - S_min_VAL
OVERFLOW_ADDRESS.assigned_stake() = O_A_S + S_min_VAL

Fig. 8

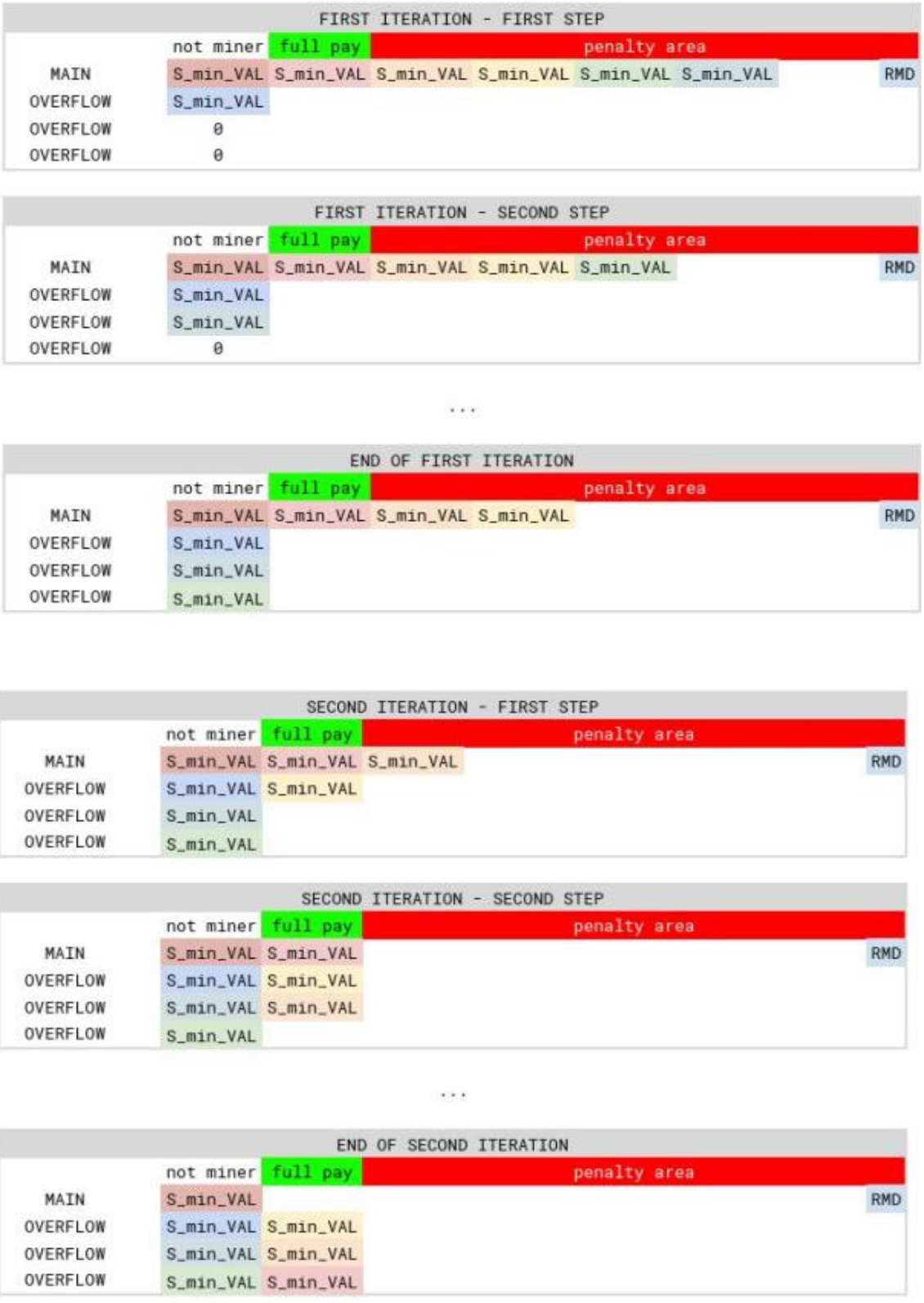
FIRST ITERATION - FIRST STEP
not miner
full pay
penalty area
MAIN
S_min_VAL S_min_VAL S_min_VAL S_min_VAL S_min_VAL S_min_VAL
RMD
OVERFLOW
S_min_VAL
OVERFLOW
0
OVERFLOW
0
FIRST ITERATION - SECOND STEP
not miner
full pay
penalty area
MAIN
S_min_VAL S_min_VAL S_min_VAL S_min_VAL S_min_VAL
RMD
OVERFLOW
S_min_VAL
OVERFLOW
S_min_VAL
OVERFLOW
0
...
END OF FIRST ITERATION
not miner
full pay
penalty area
MAIN
S_min_VAL S_min_VAL S_min_VAL S_min_VAL
RMD
OVERFLOW
S_min_VAL
OVERFLOW
S_min_VAL
OVERFLOW
S_min_VAL
SECOND ITERATION - FIRST STEP
not miner
full pay
penalty area
MAIN
S_min_VAL S_min_VAL S_min_VAL
RMD
OVERFLOW
S_min_VAL S_min_VAL
OVERFLOW
S_min_VAL
OVERFLOW
S_min_VAL
SECOND ITERATION - SECOND STEP
not miner
full pay
penalty area
MAIN
S_min_VAL S_min_VAL
RMD
OVERFLOW
S_min_VAL S_min_VAL
OVERFLOW
S_min_VAL S_min_VAL
OVERFLOW
S_min_VAL
...
END OF SECOND ITERATION
not miner
full pay
penalty area
MAIN
S_min_VAL
RMD
OVERFLOW
S_min_VAL S_min_VAL
OVERFLOW
S_min_VAL S_min_VAL
OVERFLOW
S_min_VAL S_min_VAL

Fig. 9

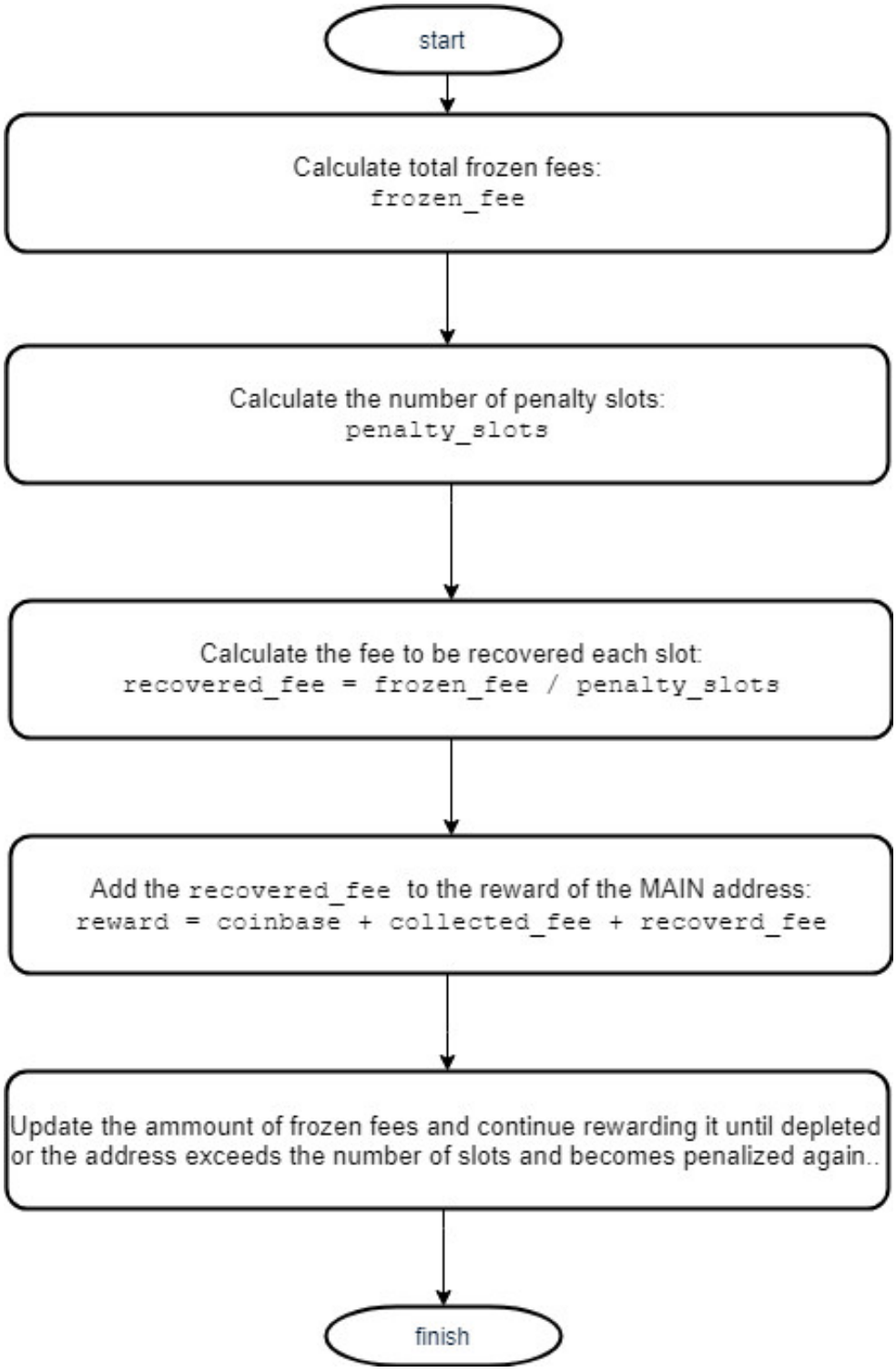
start
Calculate total frozen fees:
frozen_fee
Calculate the number of penalty slots:
penalty_slots
Calculate the fee to be recovered each slot:
recovered_fee = frozen_fee / penalty_slots
Add the recovered_fee to the reward of the MAIN address:
reward = coinbase + collected_fee + recoverd_fee
Update the ammount of frozen fees and continue rewarding it until depleted or the address exceeds the number of slots and becomes penalized again..
finish

Fig. 10

N1	ID1	2
N2	ID2	1
N3	ID3	0
N4	ID4	3
N5	ID5	0

Fig. 11

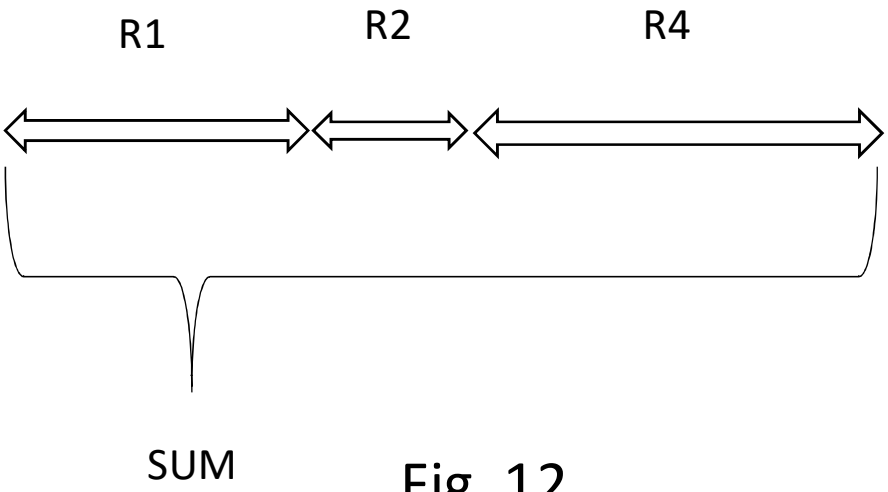

Fig. 12

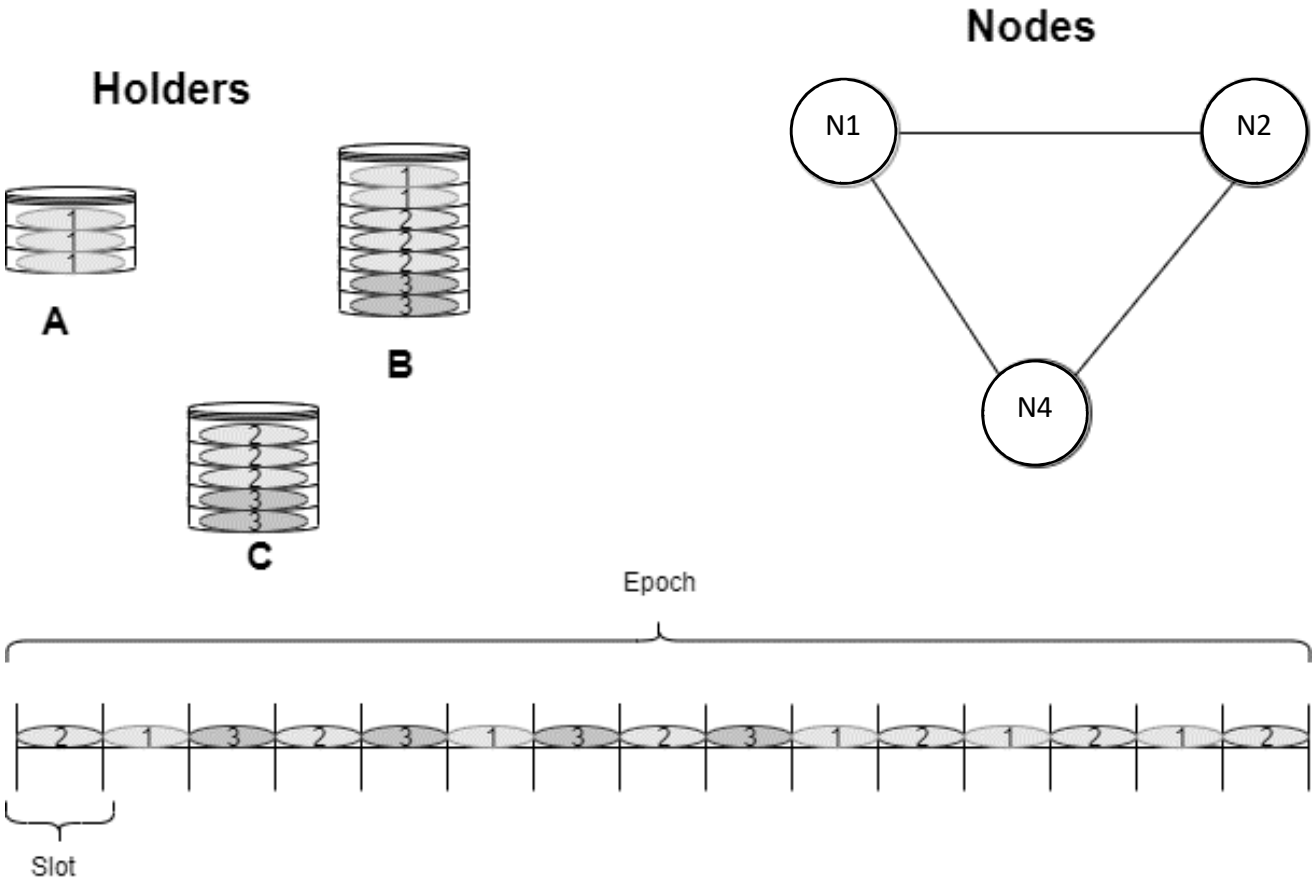

Fig. 13

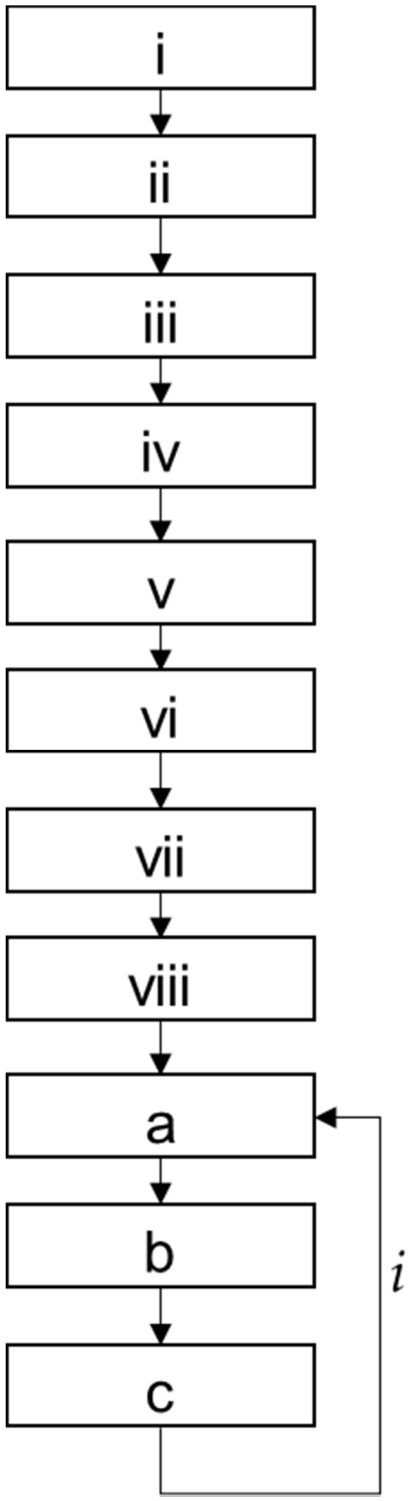

Fig. 14