

明 細 書

発明の名称：

ブロックチェーンネットワークの構成方法及びその方法を実施するための
コンピュータソフトウェアプログラム

技術分野

[0001] 本発明は、ブロックチェーンネットワークの構成方法に関するものである。

背景技術

[0002] ブロックチェーンネットワークは、多くのコンピュータやスマートフォンがノードとして参加することで、ネットワーク全体の安全性や信頼性が高まる、自立分散システムであり、データの改ざんや不正を許さないシステムといえる。

[0003] ここで、ブロックチェーンネットワークを健全に保つためには、ネットワークに参加しているノードが常にネットワークに接続し、適切に稼働している状態が望ましいが、それが難しいのが現状といえる。

[0004] すなわち、例えば、スマートフォンなどの小型携帯端末が接続可能なネットワークは、一般的にWi-Fiやモバイルネットワークとなり、データセンターにホスティングされるようなコンピュータが利用するネットワーク回線とは接続安定性やトラフィック性能などで大きく劣る場合が多い。このため、頻繁に端末がネットワークから切断されたり、再度接続されなれたりということが発生することが考えられる。

[0005] また、このような小型携帯端末において、ゲームや動画閲覧など、プロセッサの処理能力やメモリを著しく消費するアプリケーションを実行した場合、ブロックチェーンアプリケーションに割り当てられるリソースが枯渇し、動作を阻害することも考えられる。

[0006] このような状況の下、役割を果たさなくなったノードを、ブロックチェーンネットワークの一員として維持しつづけるのにはデメリットが発

生することが想定され、将来、このデメリットを最小限にしていくための方法を実装する必要がある。

発明の概要

発明が解決しようとする課題

[0007] この発明は、上記したような事情に鑑みてなされたものであり、ブロックチェーンを構成するノードの管理を適切に行え、いくつかのノードが長期的に役割を停止するような状況においても、ブロックチェーンを停止することなく、合意形成（コンセンサス）を実行し続けることができるシステム及びその方法を提供することを目的とする。

課題を解決するための手段

[0008] 上記課題を解決するために、本発明の主要な観点によれば、以下の発明が提供される。

[0009] （１） ブロックチェーンネットワークの構成方法であり、
コンピュータが、P2Pネットワーク経由で受け取ったデータをブロックチェーンに追加するための合意形成に関わるノードの情報を保持し、コンセンサスアルゴリズムに基づき、署名すべきノードを決定するノード管理工程を有し、

このノード管理工程は、

コンピュータが、合意形成に参加する全てのノードを管理し、これらのノード環状に接続されてなる仮想ネットワークを構築するものであり、

コンピュータが、前記仮想ネットワークを構成するノードのうち、どのノードがブロックを署名できたかを記録する署名ノード記録工程と、

前記署名ノード記録工程での記録に基づき、一定閾値回数連続してブロック署名に関与しなかったノードを判定し、前記仮想ネットワークを構成するノードから削除するペナルティ工程と

を有することを特徴とする方法。

[0010] （２） 前記（１）記載の方法において、

前記ペナルティ工程は、

コンピュータが、所定のタイミングで、前記仮想ネットワークを構成するノードのうち、所定のノードが署名できたかを確認し、署名できていない場合、署名していないことを示すカウンタ値をインクリメントする署名不能ノード記録工程と、

コンピュータが、所定のタイミングで、前記仮想ネットワークを構成する各ノードの前記署名していないことを示すカウンタ値が所定の閾値を超えているかを判定し、超えているノードがあると判定した場合、当該ノードを前記仮想ネットワークを構成するノードから削除するノード削除工程と、

を有することを特徴とする方法。

[0011] (3) 前記(1)記載の方法において、

前記所定のタイミングは、ブロックが生成されるタイミングであることを特徴とする方法。

[0012] (4) 前記(1)記載の方法において、

前記署名ノード記録工程は、ブロックが生成されるたびに、そのブロックを生成したノードのブロック署名カウンタのカウンタ値をインクリメントするものであり、そのブロック署名カウンタの値によりどのノードがブロックを署名できたかを記録するものである、

ことを特徴とする方法。

[0013] (5) 前記(4)記載の方法において、

前記署名不能ノード記録工程は、確認の際に、当該所定のノードが署名不能カウンタ値から署名カウンタ値を減算することで署名不能カウンタ値を調整する工程をさらに有するものである

ことを特徴とする方法。

[0014] (6) 前記(1)記載の方法において、

このノード管理工程は、

コンピュータが、合意形成に参加するノードを、より安定した環境で動作するVoterノードと、それ以外のSignerノードに定義・区別して管理し、これらのノードを互いに接続し、環状の仮想ネットワークを構築す

るものであり、

前記ペナルティ工程は、ノードがV o t e rノードであるかを判定し、V o t e rノードである場合には、前記仮想ネットワークを構成するノードから削除しないものである

ことを特徴とする方法。

[0015] (7) ブロックチェーンネットワーク構成用のコンピュータソフトウェアプログラムであり、

コンピュータに、P 2 Pネットワーク経由で受け取ったデータをブロックチェーンに追加するための合意形成に関わるノードの情報を保持し、コンセンサスアルゴリズムに基づき、署名すべきノードを決定するノード管理工程を実行させるものであり、

このノード管理工程は、

コンピュータが、合意形成に参加する全てのノードを管理し、これらのノード環状に接続されてなる仮想ネットワークを構築するものであり、

コンピュータが、前記仮想ネットワークを構成するノードのうち、どのノードがブロックを署名できたかを記録する署名ノード記録工程と、

前記署名ノード記録工程での記録に基づき、一定閾値回数連続してブロック署名に関与しなかったノードを判定し、前記仮想ネットワークを構成するノードから削除するペナルティ工程と

を実行するものである

ことを特徴とするコンピュータソフトウェアプログラム。

[0016] (8) 前記(7)記載のコンピュータソフトウェアプログラムにおいて、

前記ペナルティ工程は、

コンピュータが、所定のタイミングで、前記仮想ネットワークを構成するノードのうち、所定のノードが署名できたかを確認し、署名できていない場合、署名していないことを示すカウンタ値をインクリメントする署名不能ノード記録工程と、

コンピュータが、所定のタイミングで、前記仮想ネットワークを構成する各ノードの前記署名していないことを示すカウンタ値が所定の閾値を超えているかを判定し、超えているノードがあると判定した場合、当該ノードを前記仮想ネットワークを構成するノードから削除するノード削除工程と、
を実行するものであることを特徴とするコンピュータソフトウェアプログラム。

[0017] (9) 前記(7)記載のコンピュータソフトウェアプログラムにおいて、

前記所定のタイミングは、ブロックが生成されるタイミングであることを特徴とするコンピュータソフトウェアプログラム。

[0018] (10) 前記(7)記載のコンピュータソフトウェアプログラムにおいて、

前記署名ノード記録工程は、ブロックが生成されるたびに、そのブロックを生成したノードのブロック署名カウンタのカウンタ値をインクリメントするものであり、そのブロック署名カウンタの値によりどのノードがブロックを署名できたかを記録するものである、

ことを特徴とするコンピュータソフトウェアプログラム。

[0019] (11) 前記(10)記載のコンピュータソフトウェアプログラムにおいて、

前記署名不能ノード記録工程は、確認の際に、当該所定のノードが署名不能カウンタ値から署名カウンタ値を減算することで署名不能カウンタ値を調整する工程をさらに有するものである

ことを特徴とするコンピュータソフトウェアプログラム。

[0020] (12) 前記(7)記載のコンピュータソフトウェアプログラムにおいて、

このノード管理工程は、

コンピュータが、合意形成に参加するノードを、より安定した環境で動作するVoterノードと、それ以外のSignerノードに定義・区別して

管理し、これらのノードを互いに接続し、環状の仮想ネットワークを構築するものであり、

前記ペナルティ工程は、ノードがV o t e rノードであるかを判定し、V o t e rノードである場合には、前記仮想ネットワークを構成するノードから削除しないものである

ことを特徴とするコンピュータソフトウェアプログラム。

[0021] なお、上記した以外の本発明の特徴は、以下に説明する本発明の実施形態の説明及びその図面に開示されている。

図面の簡単な説明

[0022] [図1]図1は、本発明の一実施形態を示すシステム概略構成図である。

[図2]図2は、同じく、S e a l e r R i n gを示す模式図である。

[図3]図3は、同じく、いくつかのS e a l e rが停止した状態を示す模式図である。

[図4]図4は、同じく、ブロック署名確認の処理ロジックを示すフローチャートである。

[図5]図5は、同じく、ペナルティ処理実行時の処理ロジックを示すフローチャートである。

発明を実施するための形態

[0023] 以下、添付した図面を参照して本発明の一実施形態を説明する。

[0024] (一実施形態)

図1は、本発明の一実施形態であるブロックチェーン実行サーバモジュール(コンピュータソフトウェアプログラム)1の構成を示すものである。

[0025] この実行サーバモジュール1は、ブロックチェーンネットワークを構築するためのサービスプログラムであり、L i n u x (登録商標)などのサーバマシンで動作しても良いし、アプリケーションプログラムに組み込んで携帯端末で動作しても良い。その場合、このモジュール1は、それぞれのデバイス上に実装された記憶媒体上に記憶され、デバイスのC P U (図示せず)によってメモリ (図示せず)上に適宜呼び出されて実行されることで、本発明

の各構成として動作することになる。

[0026] この実行サーバモジュール 1 は、図 1 に示すように、スマートコントラクト実行部 2 と、コンセンサスアルゴリズム実行部 3 と、ブロックチェーンデータ保持機能部 4 と、データ同期機能部 5 と、P2P ネットワーク構築機能部 6 とを有する。ここで、スマートコントラクト実行部 2 は、ブロックチェーンに対するインタフェースも提供するものである。コンセンサスアルゴリズム実行部 3 は、ノード管理機能部 7 も提供する。

[0027] これらの各構成要素 2～7 は、スマートコントラクト 10、コンセンサスアルゴリズム 11、ブロックチェーンデータ 12、ノード情報 13 を格納する共有データ 9 にアクセス可能になっている。

[0028] 以下、これらの各構成要素の詳しい構成及び機能をその動作を通じて詳細に説明する。

[0029] (スマートコントラクト実行部、ブロックチェーンデータ保持機能部、データ同期機能部、P2P ネットワーク構築機能部の機能)

スマートコントラクト実行部 2 は、ブロックチェーンネットワークに対する外部からの呼び出しや転送データに応じて、共有データ 9 に格納されたスマートコントラクト 10 の特定のメソッド（更新メソッド）を実行し、その結果を返す仮想マシンである。

[0030] P2P ネットワーク構築機能部 6 は、新しいノードをブロックチェーンの P2P ネットワークの構成員として構築し、ブロックチェーン実行サーバ同士を接続する機能を奏するものである。すなわち、スマートコントラクト 10 のメソッドを呼び出して実行するには、ブロックチェーンネットワークにアクセスするためのノードが必要である。あるノードを新たにブロックチェーンネットワークに参加させるには、既存ノードの一つがそのノードをブロックチェーンの P2P ネットワークの構成員にする必要がある。

[0031] データ同期機能部 5 は、各ノードにおいて、最新のブロックチェーンデータ 12 の同期を行い、この同期の際、P2P で接続している他のノードからデータ 11 を取り寄せ、正しさの検証を行った上で共有データ 9 として保存

するものである。ブロックチェーンデータ保持機能部4は、サーバや携帯端末のストレージエリアに上記ブロックチェーンの共有データ（ブロックチェーンデータ12）を保持するものである。

[0032] 上記スマートコントラクト10の更新メソッドの呼び出しにおいては、トランザクションを組み立て、それを各ノードからブロックチェーンネットワークに送信する。当該トランザクションはブロックチェーンネットワークを構成するすべてのノードに伝搬する。このとき、各ノードでは、ブロックへの格納待ちのトランザクションとしてひとまずキューイングする。トランザクションがブロックに格納されるまでには一定の時間がかかるため、更新メソッドは原則的に非同期処理である。

[0033] スマートコントラクト10の更新メソッドの実行は、当該メソッドの呼び出しトランザクションが新しいブロックに格納されることに相当する。更新メソッドの実行タイミングは、ブロック作成ノードで新たなブロックが作られるとき、および、ブロック作成ノードではない他のノードが新たなブロックを受信し、その正しさを検証するときである。

[0034] スマートコントラクト10はブロックチェーンネットワーク上のすべてのノードで全く同じように実行され、処理に伴う状態更新も同じ結果に到達する。

[0035] スマートコントラクト10の参照メソッドの呼び出しと実行では、ノード自身が持っている検証済みのスマートコントラクトコードと内部状態を使って即座に結果を返す。スマートコントラクト10が以上のように実行できるのは、上記のように、ブロックチェーンネットワーク上のすべてのノードがデータ9を共有しているためである。

[0036] なお、共有対象にはブロックとトランザクションはもちろんのこと、スマートコントラクトのコードと内部状態も含まれる。これにより、どのノードでもブロックに格納されたトランザクションからスマートコントラクトの実行を再生することができ、実行にともなう内部状態更新の正しさを独自検証することができる。このような仕組みにより、ブロックチェーンの耐改ざん

性などの特徴がスマートコントラクトの実行プラットフォームとして実現される。

[0037] (コンセンサスアルゴリズム実行部)

次に、この実施形態のコンセンサスアルゴリズム実行部3について説明する。

[0038] 本実施形態におけるコンセンサスアルゴリズム11は、Proof of Authority (PoA) をベースとして採用するものである。このPoAは、信頼できるノードのみがブロックを署名できるという考えに基づくコンセンサスアルゴリズムで、Sealerと呼ばれるノードが、リング状の仮想的ネットワークを形成し、指定された時間ごとに、ブロックの署名を順番に行っていくコンセンサスアルゴリズムである。

[0039] この実施形態において、このアルゴリズムを採用するネットワークに参加するには、eKYCなどを経由して承認され信頼される必要があり (Authority)、その意味において、中央集権的ではあるが、悪意の参加者を現実的なレベルで排除できるものである。

[0040] 以下、このようなコンセプトを有するコンセンサスアルゴリズム11をこの実施形態のコンセンサスアルゴリズム実行部3及びノード管理機能部に実装した例で説明する。

[0041] このコンセンサスアルゴリズム実行部3は、上記P2Pネットワーク経由で受け取ったデータをブロックチェーンに追加するための合意形成を実行するものである。このため、合意形成に関わる全てのノードの情報13を保持する。そして、このコンセンサスアルゴリズム実行部3に設けられたノード管理機能部7が、コンセンサスアルゴリズム11に基づき、署名すべきノードを管理し決定する。

[0042] 以下、このコンセンサスアルゴリズム実行部3及びノード管理機能部7による動作について説明する。

[0043] (サイナーとボーター)

上記ノード管理機能部7は、合意形成に参加するノードを、Signer(

サイナー)とVoter (ボーター)に分けて管理する。Signerは携帯端末で動作することを想定したノードで、Voterはネットワークや電源やその他の計算資源が潤沢に付与できる、データセンター上にホスティングされるような、サーバコンピュータで動作するものである。

[0044] [表1]

Voter	ブロックを署名する以外にも、ブロックチェーンネットワークに Signer を追加したり削除を行うことのできる投票権をもつ特別なノードであり、安定的な運用が担保できるデータセンターなどに構築される。
Signer	ブロックの署名のみを行うノードであり、eKYCなどの Authority を経由して、本ブロックチェーンネットワークに参加する一般的なノードである。携帯端末などでも実行される。

[0045] ノード管理機能部は、VoterとSignerとで、図2のようなリング状の仮想ネットワークを構築し、順番にトランザクションの含まれたブロックを署名し続けることで、ネットワークを動かしていくように構成する。ここで、VoterとSignerはSealer(ブロックを署名「Sealing」しブロックチェーンに追加できるノード：シーラー)と呼ばれる。この図では、S1→V1→S2→V2のように順番にブロックを署名し、ブロックチェーンに追加していくことになる。

[0046] (Sealer Ringにおけるノード管理)

この健全な状態をSealer Ringと呼ぶ。このSealer Ringから、仮にすべてのSignerがモバイルネットワークの全体的障害などにより、ブロックチェーンネットワークから切断されたとする。この場合、図3のように、Voterだけが安定的なインフラ上に構築されているため、ネットワークを実行しつづけることができる。(図3)

しかしながら、一部のSignerノードがネットワークから切断された状態が続くと、このノードをブロックチェーンネットワークの一員として維持しつづけるのにはデメリットが発生することが想定される。

[0047] そこでこの実施形態のノード管理機能部7は、この状態を検出し解消できるように構成されている。

[0048] すなわち、このノード管理機能部 7 は、仮想ネットワークを構成するノードのうちどのノードがブロックを署名できたかを記録する署名ノード記録部 16 と、この署名記録に基づき一定閾値 N 回連続してブロック署名に関与しなかったノードを判定し前記ネットワークから削除するペナルティ処理実行部 17 とを有する。

[0049] 以下、これらの構成について説明する。

[0050] (署名ノード記録部)

本実施形態において、このノード管理機能部 7 は、ブロックの署名にかかわるノードをすべてリスト (リスト 1、2) (図に 14、15 で示す) として管理する。リスト 1 は連想配列であり、キーはノードのアドレスである。なお、このリスト 1 は、ノードオブジェクトの集まりであり、このノードオブジェクトはノードの属性を変数として保持するものである。

[0051] (リスト 1)

NodeList[] = { N1, N2, N3, N4,,, }。

[0052] またリスト 2 は、ノードのアドレスを保持する配列アドレスリストを管理する。リスト 2 のキーは 0 から全ノードの数-1 までの数値である。

[0053] (リスト 2)

NodeAddrList[] = { N1_addr, N2_addr, N3_addr,,, }。

[0054] この実施形態において、ブロックの署名は合意形成リストの順番で実行されるが、リストが順次処理され、最後の要素に到達すると、再度リストの最初から処理を繰り返し、図 2 に示すように環状に処理が継続していく。このことで、順番にブロックが生成されていく。

[0055] ここで、本実施形態の署名ノード記録部は、新しいブロックをネットワークから受け取って、ブロックをブロックチェーンに追加するタイミングで、どのノードがブロックを署名できたかを記録する。

[0056] すなわち、署名ノード記録部 16 は、新規ブロックを受信したタイミングで図 4 に示す処理を実行する。

[0057] すなわち、まず、ノードがブロックの署名を行っていて、それが有効であ

ったかを判断する（ステップS 1－2、1－2）。有効な署名である場合には、ノードのブロック署名カウンタ値signedCount（ブロック生成カウンタ値）をインクリメントする（ステップS 1－3）。このsignedCount値は上述したノードオブジェクトの変数として更新され格納される。

[0058] NodeList[addr].signed_counter++

なお、署名が有効でない場合には、何もせずに署名確認処理が終了する（ステップS 1－4）。

[0059] （ペナルティ実行部）

また、ペナルティ実行部16（図1）は、署名不能ノード記録部18と、ノード削除処理部19とを有する。署名不能ノード記録部19は、新規ノードが追加されたタイミングで前記リスト1のノードのうち所定のノードが署名できたかを確認し、署名できていない場合、署名していないことを示すカウンタ値をインクリメントする機能を有する。また、ノード削除処理部は、同タイミングで、前記署名していないことを示すカウンタ値が所定の閾値を超えているかを判定し、超えているノードがあると判定した場合、当該ノードを前記仮想ネットワークを構成するノードから削除する機能を有するものである。

[0060] すなわち、新規ブロックがブロックチェーンに追加された際、ペナルティ実行部は、まず、所定のインデックスで示されるノードのブロック生成状況をチェックするためにインデックス値を決定する（ステップS 2－1）。

[0061] ここでインデックス値は、この工程で確認する対象のノードを示す、リスト1中のアドレス順にソートされたノードの先頭からの位置であり、以下のように求める。

[0062] インデックス(IDX)＝現在の署名されたブロック番号／全ノードの数のあまり。

[0063] ついで、上記インデックス値で示されるノードのsignedCountが0以上であるか否かが判定され（ステップS 2－2、2－3）、0以上でなければ、今回署名に成功しなかったものとして、署名不能カウンタ値strikeCountをイン

クリメントする（ステップS 2－4）。このstrikeCount値も、上述したノードオブジェクトの変数として更新され格納される。

[0064] addr = NodeAddrList[IDX]

 NodeList[addr].strike_counter++。

[0065] 次に、そのノードがV o t e r以外でstrikeCount値が所定の閾値Nを超えたかが判定され（ステップS 2－5、S 2－6）、超えた場合にはそのノードは機能を停止しているとしてノードのリスト1から削除される（ステップS 2－7）。

[0066] addr = NodeAddrList[IDX]

 NodeList.delete(addr)。

[0067] ここで上記カウンタはノード毎に存在し、新しいブロックがブロックチェーンに追加されるたびに、チェックされる。チェックは、指定される閾値N(threshold)を超えたかどうかで判断する。

[0068] addr = NodeAddrList[IDX]

 NodeList[addr].strike_counter >= threshold(N)。

[0069] 一方、上記ステップS 2－3でインデックス位置のノードのsignedCountが0以上の場合には、そのノードのstrikeCount値がsignedCount値よりも大きいか否かが判定され（ステップS 2－8、2－9）、大きい場合には、strikeCount値からsignedCount値を引いた値をstrikeCount値として更新し（ステップS 2－10）、小さい場合には、strikeCount値を0に戻して初期化する（ステップS 2－11）。

[0070] addr = NodeAddrList[IDX]

 NodeList[addr].strike_counter -= NodeList[addr].signed_counter

 If(NodeList[addr].strike_counter < 0) then NodeList[addr].strike_counter = 0

 NodeList[addr].signed_counter = 0。

[0071] 以上の処理によれば、一定の閾値N回)連続して、ブロック署名に関与しなかったノードは、合意形成のリストより削除される。削除されたノードは、

合意形成に関与しなくなり、その分の処理負荷が当該のブロックチェーンネットワーク全体より除外される。

[0072] なお、ここで閾値Nは、1以上であるが、整数である必要はない。また、この閾値の決め方として、例えば、特定のノードがある日数オフラインだったらペナルティとしてネットワークから切断するというような決めの場合、以下ようになる。

すなわち、例えば、ブロック生成タイミングが5分で、構成ノードの数が2000個の仮想ネットワークの場合に、30日オフラインの場合に、ペナルティを与えるとすると、 $(30 \times 86400) / 300 / 2000 = 4.32$ となるので、おおよそ、4回署名できなければオフラインと見ることができる。個の場合、前記閾値は、4ないし4.3等と設定することができる。

[0073] また、上記インデックス値の決め方は上記一実施形態のものに限定されるものではなく、要は、各ネットワークを構成するノードの状態を順にチェックしていくことができるアルゴリズムであれば良い。例えば、上記一実施形態ではインデックス値を決定するのにブロック番号を用いたが、単にリスト上のノードの状態を順にチェックしていく方法であっても良い。

[0074] なお、本発明は上記一実施形態に限定されるものではなく、発明の要旨を変更しない範囲で他にも種々変形可能である。

符号の説明

- [0075] 1…ブロックチェーン実行サーバモジュール
3…コンセンサスアルゴリズム実行部
4…ブロックチェーンデータ保持機能部
5…データ同期機能部
6…ネットワーク構築機能部
7…ノード管理機能部
9…共有データ
10…スマートコントラクト

- 1 1 …コンセンサスアルゴリズム
- 1 2 …ブロックチェーンデータ
- 1 3 …ノード情報
- 1 6 …署名ノード記録部
- 1 7 …ペナルティ実行処理部

請求の範囲

[請求項1]

ブロックチェーンネットワークの構成方法であり、

コンピュータが、P2Pネットワーク経由で受け取ったデータをブロックチェーンに追加するための合意形成に関わるノードの情報を保持し、コンセンサスアルゴリズムに基づき、署名すべきノードを決定するノード管理工程を有し、

このノード管理工程は、

コンピュータが、合意形成に参加する全てのノードを管理し、これらのノード環状に接続されてなる仮想ネットワークを構築するものであり、

コンピュータが、前記仮想ネットワークを構成するノードのうち、どのノードがブロックを署名できたかを記録する署名ノード記録工程と、

前記署名ノード記録工程での記録に基づき、一定閾値回数連続してブロック署名に関与しなかったノードを判定し、前記仮想ネットワークを構成するノードから削除するペナルティ工程と

を有することを特徴とする方法。

[請求項2]

請求項1記載の方法において、

前記ペナルティ工程は、

コンピュータが、所定のタイミングで、前記仮想ネットワークを構成するノードのうち、所定のノードが署名できたかを確認し、署名できていない場合、署名していないことを示すカウンタ値をインクリメントする署名不能ノード記録工程と、

コンピュータが、所定のタイミングで、前記仮想ネットワークを構成する各ノードの前記署名していないことを示すカウンタ値が所定の閾値を超えているかを判定し、超えているノードがあると判定した場合、当該ノードを前記仮想ネットワークを構成するノードから削除するノード削除工程と、

を有することを特徴とする方法。

[請求項3]

請求項1記載の方法において、

前記所定のタイミングは、ブロックが生成されるタイミングであることを特徴とする方法。

[請求項4]

請求項1記載の方法において、

前記署名ノード記録工程は、ブロックが生成されるたびに、そのブロックを生成したノードのブロック署名カウンタのカウント値をインクリメントするものであり、そのブロック署名カウンタの値によりどのノードがブロックを署名できたかを記録するものである、
ことを特徴とする方法。

[請求項5]

請求項4記載の方法において、

前記署名不能ノード記録工程は、確認の際に、当該所定のノードが署名不能カウンタ値から署名カウンタ値を減算することで署名不能カウンタ値を調整する工程をさらに有するものである
ことを特徴とする方法。

[請求項6]

請求項1記載の方法において、

このノード管理工程は、
コンピュータが、合意形成に参加するノードを、より安定した環境で動作するV o t e rノードと、それ以外のS i g n e rノードに定義・区別して管理し、これらのノードを互いに接続し、環状の仮想ネットワークを構築するものであり、
前記ペナルティ工程は、ノードがV o t e rノードであるかを判定し、V o t e rノードである場合には、前記仮想ネットワークを構成するノードから削除しないものである
ことを特徴とする方法。

[請求項7]

ブロックチェーンネットワーク構成用のコンピュータソフトウェアプログラムであり、
コンピュータに、P 2 Pネットワーク経由で受け取ったデータをブ

ロックチェーンに追加するための合意形成に関わるノードの情報を保持し、コンセンサスアルゴリズムに基づき、署名すべきノードを決定するノード管理工程を実行させるものであり、

このノード管理工程は、

コンピュータが、合意形成に参加する全てのノードを管理し、これらのノード環状に接続されてなる仮想ネットワークを構築するものであり、

コンピュータが、前記仮想ネットワークを構成するノードのうち、どのノードがブロックを署名できたかを記録する署名ノード記録工程と、

前記署名ノード記録工程での記録に基づき、一定閾値回数連続してブロック署名に関与しなかったノードを判定し、前記仮想ネットワークを構成するノードから削除するペナルティ工程と

を実行するものである

ことを特徴とするコンピュータソフトウェアプログラム。

[請求項8]

請求項7記載のコンピュータソフトウェアプログラムにおいて、

前記ペナルティ工程は、

コンピュータが、所定のタイミングで、前記仮想ネットワークを構成するノードのうち、所定のノードが署名できたかを確認し、署名できていない場合、署名していないことを示すカウンタ値をインクリメントする署名不能ノード記録工程と、

コンピュータが、所定のタイミングで、前記仮想ネットワークを構成する各ノードの前記署名していないことを示すカウンタ値が所定の閾値を超えているかを判定し、超えているノードがあると判定した場合、当該ノードを前記仮想ネットワークを構成するノードから削除するノード削除工程と、

を実行するものであることを特徴とするコンピュータソフトウェアプログラム。

- [請求項9] 請求項7記載のコンピュータソフトウェアプログラムにおいて、
前記所定のタイミングは、ブロックが生成されるタイミングであることを特徴とするコンピュータソフトウェアプログラム。
- [請求項10] 請求項7記載のコンピュータソフトウェアプログラムにおいて、
前記署名ノード記録工程は、ブロックが生成されるたびに、そのブロックを生成したノードのブロック署名カウンタのカウント値をインクリメントするものであり、そのブロック署名カウンタの値によりどのノードがブロックを署名できたかを記録するものである、
ことを特徴とするコンピュータソフトウェアプログラム。
- [請求項11] 請求項10記載のコンピュータソフトウェアプログラムにおいて、
前記署名不能ノード記録工程は、確認の際に、当該所定のノードが署名不能カウンタ値から署名カウンタ値を減算することで署名不能カウンタ値を調整する工程をさらに有するものである
ことを特徴とするコンピュータソフトウェアプログラム。
- [請求項12] 請求項7記載のコンピュータソフトウェアプログラムにおいて、
このノード管理工程は、
コンピュータが、合意形成に参加するノードを、より安定した環境で動作するV o t e rノードと、それ以外のS i g n e rノードに定義・区別して管理し、これらのノードを互いに接続し、環状の仮想ネットワークを構築するものであり、
前記ペナルティ工程は、ノードがV o t e rノードであるかを判定し、V o t e rノードである場合には、前記仮想ネットワークを構成するノードから削除しないものである
ことを特徴とするコンピュータソフトウェアプログラム。