(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7575833号
(P7575833)

(45)発行日　令和6年10月30日(2024.10.30)

(24)登録日　令和6年10月22日(2024.10.22)

(51)国際特許分類
Ｈ０４Ｌ　9/32 (2006.01)

ＦＩ
Ｈ０４Ｌ　9/32　２００Ｚ

請求項の数　14　（全16頁）

(21)出願番号　特願2024-521231(P2024-521231)
(86)(22)出願日　令和5年5月29日(2023.5.29)
(86)国際出願番号　PCT/JP2023/019955
(87)国際公開番号　WO2024/004485
(87)国際公開日　令和6年1月4日(2024.1.4)
審査請求日　令和6年4月8日(2024.4.8)
(31)優先権主張番号　特願2022-105620(P2022-105620)
(32)優先日　令和4年6月30日(2022.6.30)
(33)優先権主張国・地域又は機関
日本国(JP)
早期審査対象出願

(73)特許権者　522263426
株式会社ＣｏｕｎｔＵｐ
東京都目黒区青葉台１丁目６－５４
(74)代理人　100104411
弁理士　矢口 太郎
(72)発明者　パイェク ヤクブ
東京都目黒区青葉台１丁目６－５４ 株式会社ＣｏｕｎｔＵｐ内
(72)発明者　石田 宏樹
東京都目黒区青葉台１丁目６－５４ 株式会社ＣｏｕｎｔＵｐ内
(72)発明者　大泉 洋
東京都目黒区青葉台１丁目６－５４ 株式会社ＣｏｕｎｔＵｐ内
審査官　行田 悦資

最終頁に続く

(54)【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エアプログラム

(57)【特許請求の範囲】
【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、署名する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、
コンピュータが、上記仮想ネットワークを構成するノードのうち一定数のノードが仮想ネットワークから切断されたかを検出し、そのことが検出された場合には、前記仮想ネットワークを、Ｖｏｔｅｒノードだけで構成されるＶｏｔｅｒ　Ｒｉｎｇ仮想ネットワークを移行させる工程
を有するものである方法、
【請求項２】
請求項１記載のブロックチェーンネットワークの構成方法において、
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を実行するコンセンサスアルゴリズム実行工程を有し、
このコンセンサスアルゴリズム実行工程は、前記仮想ネットワークが、Ｖｏｔｅｒノー

ドだけで構成されるＶｏｔｅｒ　Ｒｉｎｇ仮想ネットワークに移行した場合、前記Ｖｏｔｅｒノードだけで合意形成を実行するものである、
　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、
　前記ノード管理工程は、
　コンピュータが、ブロックの署名タイミングごとに、新規のブロックの追加が発生しなくなることを検出することで一定数のノードがネットワークから切断されたかを検出するブロック追加監視工程を有し、
　前記ブロック追加監視工程で上記のことが検出されると、一定の期間ブロックが署名されないかどうかを確認するモードに入り、その期間を過ぎても新規ブロックがブロックチェーンに追加されない場合、Ｖｏｔｅｒだけで構成される、Ｖｏｔｅｒ　Ｒｉｎｇにネットワークを移行する移行工程と
　を有す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、
　コンピュータが、ブロックに記録されるｄｉｆｆｉｃｕｌｔｙ値を変更することでＶｏｔｅｒ　Ｒｉｎｇモードによるブロック追加か、通常のブロック追加かを区別する工程を有する
　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、
　上記ｄｉｆｆｉｃｕｌｔｙ値を変更は、Ｖｏｔｅｒ　Ｒｉｎｇモードによるブロック追加の場合の値のレンジを、通常のブロック追加の場合よりも高く設定するものである
　ことを特徴とする方法。
【請求項６】
請求項４記載の方法にいて、
　ネットワークに復帰したＳｉｇｎｅｒノードは、自分の署名順になった場合に、直前のブロックがＶｏｔｅｒ　Ｒｉｎｇで書き込まれたと判断した場合、それまでのＶｏｔｅｒ　Ｒｉｎｇにおけるｄｉｆｆｉｃｕｌｔｙより高いレンジのｄｉｆｆｉｃｕｌｔｙを指定して新規のブロックを署名することで、Ｖｏｔｅｒ　Ｒｉｎｇから通常Ｒｉｎｇへの復帰を促すものである
　ことを特徴とする方法。
【請求項７】
請求項６記載の方法において、
　すべてのＳｉｇｎｅｒノードは、Ｖｏｔｅｒ　Ｒｉｎｇのｄｉｆｆｉｃｕｌｔｙを超えるｄｉｆｆｉｃｕｌｔｙを確認したことに基づき、ｄｉｆｆｉｃｕｌｔｙ値を元の値に戻しＶｏｔｅｒ　Ｒｉｎｇの状態を解消し、全ノードの参加する通常状態に復帰させるものである
　ことを特徴とする方法。
【請求項８】
ブロックチェーンネットワークを構築及び維持するためのコンピュータソフトウエアプログラムであり、
　コンピュータに、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、署名するべきノードを決定させ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、

コンピュータが、上記仮想ネットワークを構成するノードのうち一定数のノードが仮想ネットワークから切断されたかを検出し、そのことが検出された場合には、前記仮想ネットワークを、Ｖｏｔｅｒノードだけで構成されるＶｏｔｅｒ　Ｒｉｎｇ仮想ネットワークを移行させる工程
を有するものである
コンピュータソフトウエアプログラム。
【請求項９】
請求項８記載のコンピュータソフトウエアプログラムにおいて、
コンピュータに、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させるコンセンサスアルゴリズム実行工程を有し、
このコンセンサスアルゴリズム実行工程は、前記仮想ネットワークが、Ｖｏｔｅｒノードだけで構成されるＶｏｔｅｒ　Ｒｉｎｇ仮想ネットワークに移行した場合、前記Ｖｏｔｅｒノードだけで合意形成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エアプログラム。
【請求項１０】
請求項８記載の方法において、
前記ノード管理工程は、
コンピュータが、ブロックの署名タイミングごとに、新規のブロックの追加が発生しなくなることを検出することで一定数のノードがネットワークから切断されたかを検出するブロック追加監視工程を有し、
前記ブロック追加監視工程で上記のことが検出されると、一定の期間ブロックが署名されないかどうかを確認するモードに入り、その期間を過ぎても新規ブロックがブロックチェーンに追加されない場合、Ｖｏｔｅｒだけで構成される、Ｖｏｔｅｒ　Ｒｉｎｇにネットワークを移行する移行工程と
を有することを特徴とするコンピュータソフトウエアプログラム。
【請求項１１】
請求項８記載の方法において、
コンピュータが、ブロックに記録されるｄｉｆｆｉｃｕｌｔｙ値を変更することでＶｏｔｅｒ　Ｒｉｎｇモードによるブロック追加か、通常のブロック追加かを区別する工程を有する
ことを特徴とするコンピュータソフトウエアプログラム。
【請求項１２】
請求項１１記載の方法において、
上記ｄｉｆｆｉｃｕｌｔｙ値を変更は、Ｖｏｔｅｒ　Ｒｉｎｇモードによるブロック追加の場合の値のレンジを、通常のブロック追加の場合よりも高く設定するものである
ことを特徴とする方法。
【請求項１３】
請求項１１記載のコンピュータソフトウエアプログラムにおいて、
ネットワークに復帰したＳｉｇｎｅｒノードは、自分の署名順になった場合に、直前のブロックがＶｏｔｅｒ　Ｒｉｎｇで書き込まれたと判断した場合、それまでのＶｏｔｅｒ　Ｒｉｎｇにおけるｄｉｆｆｉｃｕｌｔｙより高いレンジのｄｉｆｆｉｃｕｌｔｙを指定して新規のブロックを署名することで、Ｖｏｔｅｒ　Ｒｉｎｇから通常Ｒｉｎｇへの復帰を促すものである
ことを特徴とするコンピュータソフトウエアプログラム。
【請求項１４】
請求項１３記載のコンピュータソフトウエアプログラムにおいて、
すべてのＳｉｇｎｅｒノードは、Ｖｏｔｅｒ　Ｒｉｎｇのｄｉｆｆｉｃｕｌｔｙを超えるｄｉｆｆｉｃｕｌｔｙを確認したら、ｄｉｆｆｉｃｕｌｔｙ値を元の値に戻しＶｏｔｅｒ　Ｒｉｎｇの状態を解消し、全ノードの参加する通常状態に復帰させるものである
ことを特徴とするコンピュータソフトウエアプログラム。

【発明の詳細な説明】
【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ＷｉＦｉ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高性能サーバ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、再度接続されなおし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ー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、仮に、半分以上のノードがネットワークから切断されるような状況が発生すると、そのブロックチェーンネットワークは新しいブロックを追加するこができなくなり、事実上機能を停止してしまう。
【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、多くのノードが停止または切断され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。
（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、署名する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、
コンピュータが、上記仮想ネットワークを構成するノードのうち一定数のノードが仮想ネットワークから切断されたかを検出し、そのことが検出された場合には、前記仮想ネットワークを、Ｖｏｔｅｒノードだけで構成されるＶｏｔｅｒ　Ｒｉｎｇ仮想ネットワークを移行させる工程
を有するものである方法。

【０００９】
（２） 前記（１）記載のブロックチェーンネットワークの構成方法において、
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を実行するこのコンセンサスアルゴリズム実行工程を有し、
このコンセンサスアルゴリズム実行工程は、前記仮想ネットワークが、Ｖｏｔｅｒノードだけで構成されるＶｏｔｅｒ Ｒｉｎｇ仮想ネットワークに移行した場合、前記Ｖｏｔｅｒノードだけで合意形成を実行するものである、
ことを特徴とする方法。
【００１０】
（３） 前記（１）記載の方法において、
前記管理機能工程は、
コンピュータが、ブロックの署名タイミングごとに、新規のブロックの追加が発生しなくなることを検出することで一定数のノードがネットワークから切断されたかを検出するブロック追加監視工程を有し、
前記ブロック追加監視工程で上記のことが検出されると、一定の期間ブロックが署名されないかどうかを確認するモードに入り。その期間を過ぎても新規ブロックがブロックチェーンに追加されない場合、Ｖｏｔｅｒだけで構成される、Ｖｏｔｅｒ Ｒｉｎｇにネットワークを移行する移行工程と
を有することを特徴とする方法。
（４） 前記（１）記載の方法において、
コンピュータが、ブロックに記録されるｄｉｆｆｉｃｕｌｔｙ値を変更することでＶｏｔｅｒ Ｒｉｎｇモードによるブロック追加か、通常のブロック追加かを区別する工程を有する
ことを特徴とする方法。
【００１１】
（５） 前記（４）記載の方法において、
上記ｄｉｆｆｉｃｕｌｔｙ値を変更は、Ｖｏｔｅｒ Ｒｉｎｇモードによるブロック追加の場合の値のレンジを、通常のブロック追加の場合よりも高く設定するものである
ことを特徴とする方法。
【００１２】
（６） 前記（４）記載の方法にいて、
ネットワークに復帰したＳｉｇｎｅｒノードは、自分の署名順になった場合に、直前のブロックがＶｏｔｅｒ Ｒｉｎｇで書き込まれたと判断した場合、それまでのＶｏｔｅｒ Ｒｉｎｇにおけるｄｉｆｆｉｃｕｌｔｙより高いレンジのｄｉｆｆｉｃｕｌｔｙを指定して新規のブロックを署名することで、Ｖｏｔｅｒ Ｒｉｎｇから通常Ｒｉｎｇへの復帰を促すものである
ことを特徴とする方法。
【００１３】
（７） 前記（６）記載の方法において、
すべてのＳｉｇｎｅｒノードは、Ｖｏｔｅｒ Ｒｉｎｇのｄｉｆｆｉｃｕｌｔｙを超えるｄｉｆｆｉｃｕｌｔｙを確認したことに基づき、ｄｉｆｆｉｃｕｌｔｙ値を元の値に戻しＶｏｔｅｒ Ｒｉｎｇの状態を解消し、全ノードの参加する通常状態に復帰させるものである
ことを特徴とする方法。
【００１４】
（８） ブロックチェーンネットワークを構築及び維持するためのコンピュータソフトウエアプログラムであり、
コンピュータに、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、署名するべきノードを決定させ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、
コンピュータが、上記仮想ネットワークを構成するノードのうち一定数のノードが仮想ネットワークから切断されたかを検出し、そのことが検出された場合には、前記仮想ネットワークを、Ｖｏｔｅｒノードだけで構成されるＶｏｔｅｒ　Ｒｉｎｇ仮想ネットワークを移行させる工程
を有するものである
コンピュータソフトウエアプログラム。

【００１５】
（９）　前記（８）記載のコンピュータソフトウエアプログラムにおいて、
コンピュータに、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させるコンセンサスアルゴリズム実行工程を有し、
このコンセンサスアルゴリズム実行工程は、前記仮想ネットワークが、Ｖｏｔｅｒノードだけで構成されるＶｏｔｅｒ　Ｒｉｎｇ仮想ネットワークに移行した場合、前記Ｖｏｔｅｒノードだけで合意形成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エアプログラム。

【００１６】
（１０）　前記（８）記載の方法において、
前記管理機能工程は、
コンピュータが、ブロックの署名タイミングごとに、新規のブロックの追加が発生しなくなることを検出することで一定数のノードがネットワークから切断されたかを検出するブロック追加監視工程を有し、
前記ブロック追加監視工程で上記のことが検出されると、一定の期間ブロックが署名されないかどうかを確認するモードに入り。その期間を過ぎても新規ブロックがブロックチェーンに追加されない場合、Ｖｏｔｅｒだけで構成される、Ｖｏｔｅｒ　Ｒｉｎｇにネットワークを移行する移行工程と
を有することを特徴とするコンピュータソフトウエアプログラム。

【００１７】
（１１）　前記（８）記載の方法において、
コンピュータが、ブロックに記録されるｄｉｆｆｉｃｕｌｔｙ値を変更することでＶｏｔｅｒ　Ｒｉｎｇモードによるブロック追加か、通常のブロック追加かを区別する工程を有する
ことを特徴とするコンピュータソフトウエアプログラム。

【００１８】
（１２）　前記（１１）記載の方法において、
上記ｄｉｆｆｉｃｕｌｔｙ値を変更は、Ｖｏｔｅｒ　Ｒｉｎｇモードによるブロック追加の場合の値のレンジを、通常のブロック追加の場合よりも高く設定するものである
ことを特徴とする方法。

【００１９】
（１３）　前記（１１）記載のコンピュータソフトウエアプログラムにおいて、
ネットワークに復帰したＳｉｇｎｅｒノードは、自分の署名順になった場合に、直前のブロックがＶｏｔｅｒ　Ｒｉｎｇで書き込まれたと判断した場合、それまでのＶｏｔｅｒ　Ｒｉｎｇにおけるｄｉｆｆｉｃｕｌｔｙより高いレンジのｄｉｆｆｉｃｕｌｔｙを指定して新規のブロックを署名することで、Ｖｏｔｅｒ　Ｒｉｎｇから通常Ｒｉｎｇへの復帰を促すものである
ことを特徴とするコンピュータソフトウエアプログラム。

【００２０】
（１４）　前記（１３）記載のコンピュータソフトウエアプログラムにおいて、

すべてのＳｉｇｎｅｒノードは、Ｖｏｔｅｒ　Ｒｉｎｇのｄｉｆｆｉｃｕｌｔｙを超えるｄｉｆｆｉｃｕｌｔｙを確認したら、ｄｉｆｆｉｃｕｌｔｙ値を元の値に戻しＶｏｔｅｒ　Ｒｉｎｇの状態を解消し、全ノードの参加する通常状態に復帰させるものである
ことを特徴とするコンピュータソフトウエアプログラム。

【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。

【００２３】

【図２】図２は、同じく、Ｓｅａｌｅｒ　Ｒｉｎｇを示す模式図である。

【００２４】

【図３】図３は、同じく、いくつかのＳｅａｌｅｒが停止した状態を示す模式図である。

【００２５】

【図４】図４は、同じく、Ｖｏｔｅｒ　Ｒｉｎｇを示す模式図である。

【００２６】

【図５】図５は、同じく、新規ブロック生成時の処理ロジックを示すフローチャートである。

【００２７】

【図６】図６は、同じく、新規ブロック受信時の処理ロジックを示すフローチャートである。

【００２８】

【図７】図７は、同じく、Ｓｅａｌｅｒ　Ｒｉｎｇでのチェック処理を示すフローチャートである。

【００２９】

【図８】図８は、同じく、Ｖｏｔｅｒ　Ｒｉｎｇでのチェック処理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】

【００３０】

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。

【００３１】

（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１（コンピュータソフトウエアプログラム）の構成を示すものである。

【００３２】

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、Ｌｉｎｕｘ（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ＣＰＵ（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。

【００３３】

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。

【００３４】

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、ｄｉｆｆｉｃｕｌｔｙレンジ１４を格納する共有データ９にアクセス可能になっている。

【００３５】

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。

【００３６】

（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。

【００３７】

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。

【００３８】

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。

【００３９】

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。

【００４０】

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。

【００４１】

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。

【００４２】

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１１を共有しているためである。

【００４３】

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。

【００４４】
（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。
【００４５】
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。
【００４６】
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（Ａｕｒｈｏｒｉｔｙ）、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。
【００４７】
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７に実装した例で説明する。
【００４８】
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、署名するべきノードを管理し決定する。
【００４９】
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。
【００５０】
（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。
【００５１】
【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。

【００５２】
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。

【００５３】
（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）。
【００５４】
しかしながら、過半数を超えるノードがネットワークから切断されると、有効な合意形成が不可能となり、新規ブロックのブロックチェーンへの追加ができなくなるため、ブロックチェーンネットワークは停止する恐れがある。
【００５５】
オンラインに残っているＶｏｔｅｒのノード管理機能部７は、この状態を検出できるように構成されている。すなわち、ノード管理機能部７は、ブロック追加監視部１６を有し、ブロックの署名タイミングごとに、新規のブロックの追加が発生しなくなることを検出することで、このことを検出する（通常、新規ブロックの追加は全ノードが検知し、自身のブロックチェーンを更新する）。
【００５６】
ブロック追加監視部１６でこのことが検出されると、Ｖｏｔｅｒのノード管理機能部７は、一定の期間ブロックが署名されないかどうかを確認するモードに入り、その期間を過ぎても新規ブロックがブロックチェーンに追加されない場合、Ｖｏｔｅｒだけで構成される、Ｖｏｔｅｒ　Ｒｉｎｇにネットワークを移行する。
【００５７】
（Ｖｏｔｅｒ　Ｒｉｎｇの形成ロジック）
Ｖｏｔｅｒ　Ｒｉｎｇは、その時に有効なすべてのＳｉｇｎｅｒを排除した形で、Ｖｏｔｅｒノードのみの合意形成リング（Ｖｏｔｅｒ　Ｒｉｎｇ）を形成し（図４）、この状態で、Ｖｏｔｅｒは通常の新規ブロックの署名と追加のロジックを実行するように構成するものである。
【００５８】
この、Ｖｏｔｅｒノードのみの合意形成リング、すなわち、Ｖｏｔｅｒ　Ｒｉｎｇを形成するには、図１に１７で示すｄｉｆｆｉｃｕｌｔｙレンジ変更部が、ブロックに記録される、ｄｉｆｆｉｃｕｌｔｙレンジを表２のように変更し、Ｖｏｔｅｒ　Ｒｉｎｇによるブロック追加か、通常のブロック追加かを区別できるようにする。ｄｉｆｆｉｃｕｌｔｙとはあるレンジの範囲にある値のことである。この実施形態はノードの数によって決まるが、別の基準によって決定しても良い。
【００５９】
【表２】

通常の場合にブロックに書き込まれる difficulty の範囲	difficulty = <1,　全ノード数>
Voter Ring の場合にブロックに書き込まれる difficulty の範囲	difficulty = <全ノード数 + 1, 全ノード数 + Voter の数 >

【００６０】
（Ｓｅａｌｅｒ　Ｒｉｎｇへの復帰ロジック）
次に、ネットワークから切断されていた携帯端末（Ｓｉｇｎｅｒ）がオンラインに復帰した場合、復帰したノードは、まず最新のブロックチェーンの状態に合わせるため、データの同期を実行する。
【００６１】
同期が完了すると、Ｓｉｇｎｅｒのノード管理機能部７は現在のブロックチェーンが、

通常の状態で動作していたのか、Ｖｏｔｅｒ　Ｒｉｎｇで動作していたのかを、ブロックに含まれるｄｉｆｆｉｃｕｌｔｙを参照して検出する。

【００６２】

復帰したＳｉｇｎｅｒは、自分の署名順になった場合に、直前のブロックがＶｏｔｅｒ　Ｒｉｎｇで書き込まれたと判断した場合、それまでのＶｏｔｅｒ　Ｒｉｎｇにおけるｄｉｆｆｉｃｕｌｔｙレンジより大きいレンジのｄｉｆｆｉｃｕｌｔｙを指定して新規のブロックを署名することで、Ｖｏｔｅｒ　Ｒｉｎｇから通常Ｒｉｎｇへの復帰を促す。

【００６３】

【表３】

復帰に必要な新しい difficulty の範囲	difficulty = <全ノード数 + Voter の数 + 1， 2*全ノードの数 + Voter の数>

【００６４】

すべてのＶｏｔｅｒのノード管理機能部７は、このＶｏｔｅｒ　Ｒｉｎｇのｄｉｆｆｉｃｕｌｔｙを超えるｄｉｆｆｉｃｕｌｔｙ値を確認したら、Ｖｏｔｅｒ　Ｒｉｎｇの状態を解消し、全ノードの参加する通常状態に復帰させる。

【００６５】

これにより、すべてのＶｏｔｅｒはＶｏｔｅｒ　Ｒｉｎｇの状態を解除し、有効なＳｅａｌｅｒがすべて参加する。通常のモードであるＳｅａｌｅｒ　Ｒｉｎｇへ移行する。そして、前記ｄｉｆｆｉｃｕｌｔｙレンジ変更部１７は、ｄｉｆｆｉｃｕｌｔｙレンジを通常の状態のレンジに復帰させる。

【００６６】

（新規ブロック生成時の処理フロー）

図５～図８は、上記Ｓｅａｌｅｒ　ＲｉｎｇとＶｏｔｅｒ　Ｒｉｎｇ間の遷移ロジックの具体例を示すフローチャートを示すものである。

【００６７】

図５は、新規ブロックの生成時、すなわち、Ｓｅｌｅｒ自らがブロックを署名する場合のＤｉｆｆｕｃｕｌｔｙの決定ロジックを示すものである。

【００６８】

すなわち、新規ブロックの生成時には、過去のブロックチェーンの状態を読み込み、ｄｉｆｆｉｃｕｌｔｙを検証することでネットワークがＶｏｔｅｒ　Ｒｉｎｇで実行されていたかをチェックする（ステップＳ１－１、Ｓ１－２）。ここで、Ｖｏｔｅｒ　Ｒｉｎｇでない場合には（ステップＳ１－３）、前回のブロック生成からの経過時間をチェックし（ステップＳ１－４）、一定時間が経過している場合にはＶｏｔｅｒ　Ｒｉｎｇに切り替える判断をし（Ｓ１－５）、一定時間経過していない場合には、Ｖｏｔｅｒ　Ｒｉｎｇに切り替えずに通常のｄｉｆｆｉｃｕｌｔｙレンジに設定する（ステップＳ１－６）。

【００６９】

一方、すでにＶｏｔｅｒ　Ｒｉｎｇの場合、若しくはＶｏｔｅｒ　Ｒｉｎｇに切り替えると判断した場合には、自己ノードがＶｏｔｅｒか否かを判定し（ステップＳ１－７）、Ｖｏｔｅｒの場合には、Ｖｏｔｅｒ　Ｒｉｎｇのｄｉｆｆｉｃｕｌｔｙレンジに設定し（ステップＳ１－８）、逆に、Ｖｏｔｅｒではない場合（自己ノードがＳｉｇｎｅｒの場合）にはＶｏｔｅｒ　Ｒｉｎｇを解除するためのｄｉｆｆｉｃｕｌｔｙレンジに設定する（ステップＳ１－９）。

【００７０】

そしてこのノードは、上記で決定されたｄｉｆｆｉｃｕｌｔｙレンジで新規ブロックの追加処理を続行する（ステップＳ１－１０）。

【００７１】

（新規ブロック受信・追加時の処理フロー）

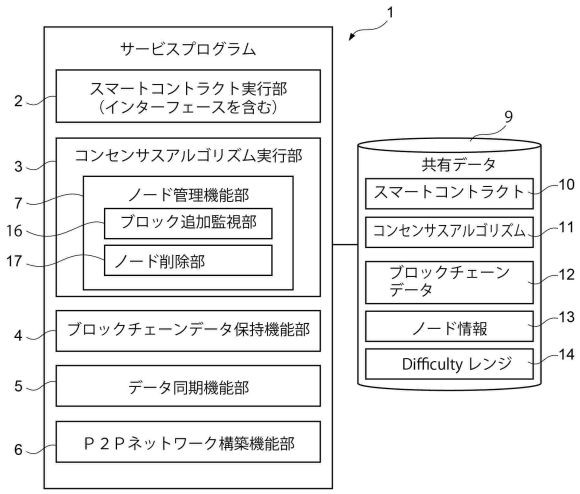
図６は、Ｓｅａｌｅｒで新規ブロックを受信した場合の新規ブロック追加ロジックを示

すものである。
【００７２】
すなわち、このロジックでは、受信したブロックの正当性をチェックする工程（ステップＳ２－１）の一部として、若しくはその後に、ステップＳ２－２以下を実行する。
【００７３】
まず、ステップＳ２－２で過去のブロックチェーンの状態を読み込んだ後、ブロックに含まれるｄｉｆｆｉｃｕｌｔｙ値に基づいてネットワークがＶｏｔｅｒ　Ｒｉｎｇで実行されているかをチェックする（ステップＳ２－３）。
【００７４】
そして、ステップＳ２－４でＶｏｔｅｒ　Ｒｉｎｇであると判定された場合には、Ｖｏｔｅｒ　Ｒｉｎｇでのチェックを実行し（ステップＳ２－５）、ステップＳ２－４でＶｏｔｅｒ　Ｒｉｎｇではないと判定された場合には、Ｓｅａｌｅｒ　Ｒｉｎｇでのチェックを実行する（ステップＳ２－６）。ステップＳ－７でチェック結果が正しい場合にはブロックを承認して保存し（ステップＳ－８）、チェック結果が正しくない場合にはエラーを返す（ステップＳ２－９）。
【００７５】
（Ｓｅａｌｅｒ　Ｒｉｎｇでのチェック実行の処理フロー）
図７は、上記新規ブロックの追加承認処理（図６）におけるステップＳ２－６で実行されるＳｅａｌｅｒ　Ｒｉｎｇでのチェック実行の処理フローを示すものである。
【００７６】
すなわち、この処理においては、ステップＳ３－１～３－３でそれぞれ、ｄｉｆｆｉｃｕｌｔｙの範囲が、Ｓｅａｌｅｒ　Ｒｉｎｇ、Ｖｏｔｅｒ　Ｒｉｎｇ，Ｖｏｔｅｒ　Ｒｉｎｇ解除用のものであるかを判定し、Ｖｏｔｅｒ　Ｒｉｎｇの範囲内にある場合には、自己ノードがＶｏｔｅｒノードであるかを判断し（Ｓ３－４）、Ｖｏｔｅｒノードでない場合にはチェック結果としてエラーを返し（Ｓ３－５）、Ｖｏｔｅｒノードの場合には、前回ブロック生成タイミングか一定時間が経過している場合のみ、肯定的なチェック結果を返す（ステップＳ３－６）。
【００７７】
一方、ステップＳ３－３でＶｏｔｅｒ　Ｒｉｎｇ解除用のｄｉｆｆｉｃｕｌｔｙ範囲内であると判断された場合には、自己ノードがＳｉｇｎｅｒノードでない場合（ステップＳ３－６で判断）にはチェック結果としてエラーを返し（ステップＳ３－７）、Ｓｉｇｎｅｒノードの場合（ステップＳ３－６で判断）には、前回ブロック生成タイミングか一定時間が経過していると判断された場合（Ｓ３－８、３－９）のみ、肯定的なチェック結果を返す（ステップＳ３－１０）。
【００７８】
（Ｖｏｔｅｒ　Ｒｉｎｇでのチェック実行の処理フロー）
図８は、上記新規ブロックの追加承認処理（図６）におけるステップＳ２－５で実行されるＶｏｔｅｒ　Ｒｉｎｇでのチェック実行の処理フローを示すものである。
【００７９】
すなわち、この処理においては、ステップＳ４－１～４－３でそれぞれ、ｄｉｆｆｉｃｕｌｔｙの範囲が、Ｓｅａｌｅｒ　Ｒｉｎｇ、Ｖｏｔｅｒ　Ｒｉｎｇ，Ｖｏｔｅｒ　Ｒｉｎｇ解除用のものであるかを判定し、Ｖｏｔｅｒ　Ｒｉｎｇの範囲内にある場合（ステップＳ４－２）には、自己ノードがＶｏｔｅｒノードであるかを判定し（ステップＳ４－４）、Ｖｏｔｅｒでない場合にはチェック結果としてエラーを返し（ステップＳ４－５）、Ｖｏｔｅｒノードの場合には、肯定的なチェック結果を返す（ステップＳ４－６）。
【００８０】
一方、Ｖｏｔｅｒ　Ｒｉｎｇ解除用のｄｉｆｆｉｃｕｌｔｙ範囲内である場合（ステップＳ４－３）には、自己ノードがＳｉｇｎｅｒノードでない場合（Ｓ４－７）にはチェック結果としてエラーを返し（ステップＳ４－８）、Ｓｉｇｎｅｒノードの場合には、肯定的なチェック結果を返す（ステップＳ４－６）。

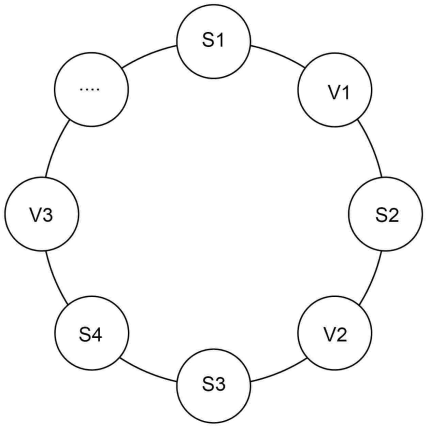
【００８１】
以上のようなノード管理機能部の処理を実装することにより、携帯端末のような比較的安定度の低いネットワークインフラを利用するノードを参加させても、ブロックチェーンネットワークを停止させることなく、安定的にブロックチェーンネットワークの動作を継続できるようになる。
【００８２】
なお、本発明は上記一実施形態に限定されるものではなく、発明の要旨を変更しない範囲で種々変形可能である。
【符号の説明】
【００８３】
１…ブロックチェーン実行サーバモジュール
３…コンセンサスアルゴリズム実行部
４…ブロックチェーンデータ保持機能部
５…データ同期機能部
６…Ｐネットワーク構築機能部
７…ノード管理機能部
９…共有データ
１０…スマートコントラクト
１１…コンセンサスアルゴリズム
１２…ブロックチェーンデータ
１３…ノード情報
１４…ｄｉｆｆｉｃｕｌｔｙレンジ
１６…ブロック追加監視部
１７…ｄｉｆｆｉｃｕｌｔｙレンジ変更部

【図面】

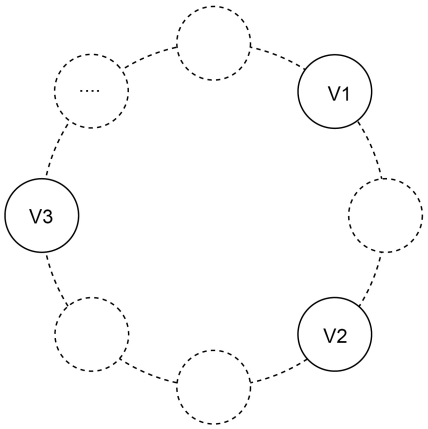
【図１】


1
サービスプログラム
2
スマートコントラクト実行部
（インターフェースを含む）
3
コンセンサスアルゴリズム実行部
7
ノード管理機能部
16
ブロック追加監視部
17
ノード削除部
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
Ｐ２Ｐネットワーク構築機能部
9
共有データ
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
Difficulty レンジ
14

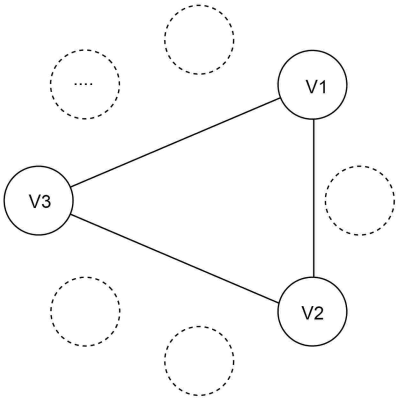
【図２】


S1
....
V1
V3
S2
S4
V2
S3

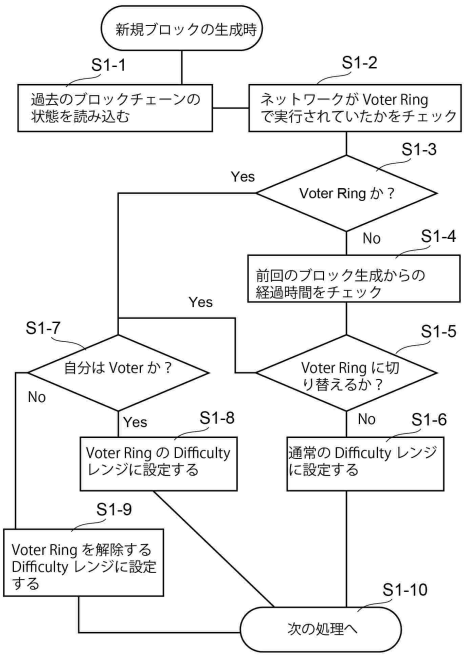
【図３】


....
V1
V3
V2

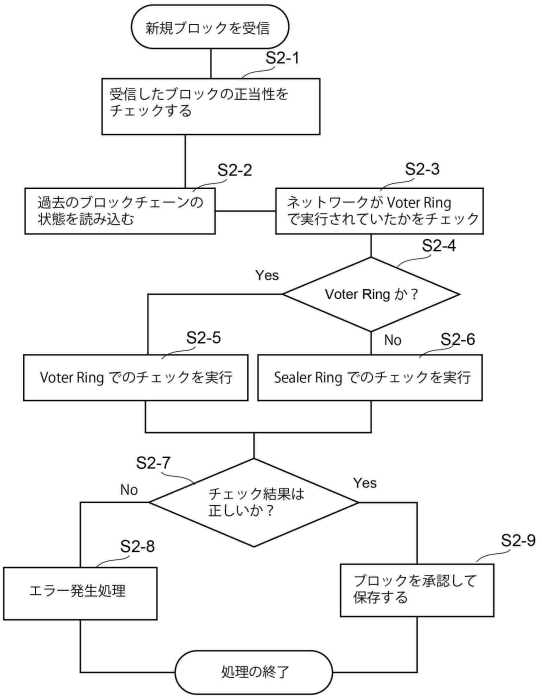
【図４】


....
V1
V3
V2

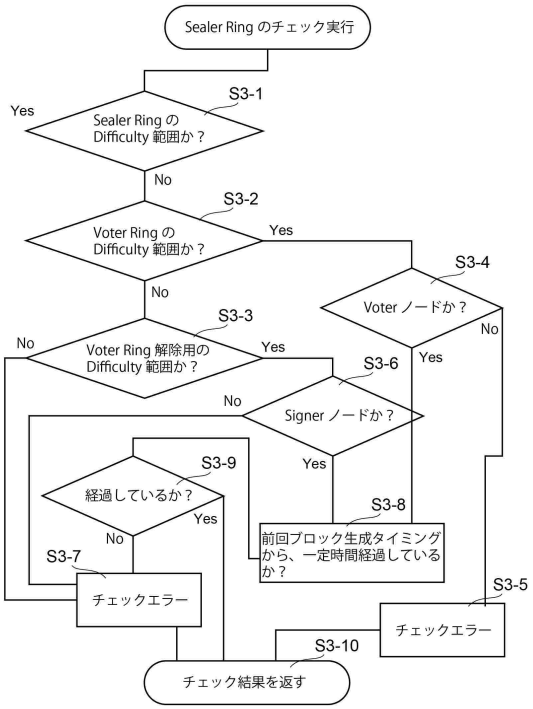
【図５】


新規ブロックの生成時
S1-1
過去のブロックチェーンの状態を読み込む
S1-2
ネットワークが Voter Ring で実行されていたかをチェック
S1-3
Voter Ring か？
Yes
No
S1-4
前回のブロック生成からの経過時間をチェック
Yes
S1-7
自分は Voter か？
No
Yes
S1-5
Voter Ring に切り替えるか？
No
Yes
S1-8
Voter Ring の Difficulty レンジに設定する
No
S1-6
通常の Difficulty レンジに設定する
S1-9
Voter Ring を解除する Difficulty レンジに設定する
S1-10
次の処理へ

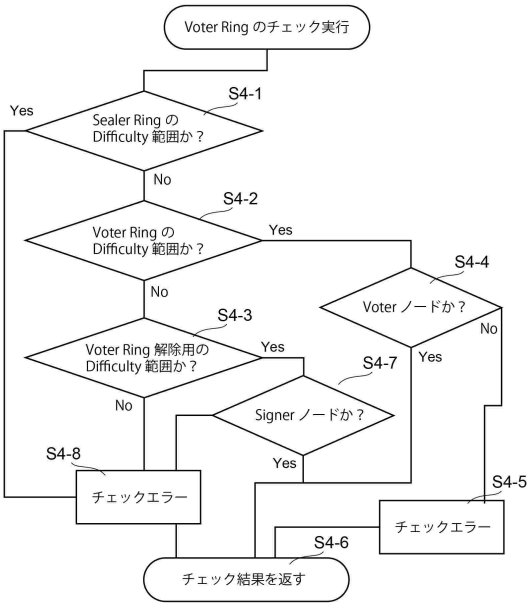
【図６】


新規ブロックを受信
S2-1
受信したブロックの正当性をチェックする
S2-2
過去のブロックチェーンの状態を読み込む
S2-3
ネットワークが Voter Ring で実行されていたかをチェック
S2-4
Voter Ring か？
Yes
No
S2-5
Voter Ring でのチェックを実行
S2-6
Sealer Ring でのチェックを実行
S2-7
チェック結果は正しいか？
No
Yes
S2-8
エラー発生処理
S2-9
ブロックを承認して保存する
処理の終了

【図７】


Sealer Ring のチェック実行
S3-1
Sealer Ring の Difficulty 範囲か？
Yes
No
S3-2
Voter Ring の Difficulty 範囲か？
Yes
No
S3-4
Voter ノードか？
No
Yes
S3-3
Voter Ring 解除用の Difficulty 範囲か？
No
Yes
S3-6
Signer ノードか？
No
Yes
S3-9
経過しているか？
Yes
No
S3-8
前回ブロック生成タイミングから、一定時間経過しているか？
S3-7
チェックエラー
S3-5
チェックエラー
S3-10
チェック結果を返す

【図８】


Voter Ring のチェック実行
S4-1
Sealer Ring の Difficulty 範囲か？
Yes
No
S4-2
Voter Ring の Difficulty 範囲か？
Yes
No
S4-4
Voter ノードか？
No
Yes
S4-3
Voter Ring 解除用の Difficulty 範囲か？
Yes
No
S4-7
Signer ノードか？
Yes
No
S4-8
チェックエラー
S4-5
チェックエラー
S4-6
チェック結果を返す

フロントページの続き

(56)参考文献 特表２０２２－５１８９６０（ＪＰ，Ａ）
国際公開第２０２０／２６３３０８（WO，Ａ１）
国際公開第２０２１／２３３０４８（WO，Ａ１）
赤羽 喜治，ブロックチェーン 仕組みと理論，日本，株式会社リックテレコム，2019年07月31日，pp.134-136

(58)調査した分野 (Int.Cl.，ＤＢ名)
Ｈ０４Ｌ ９／３２