

目次

1	POLKADOT: VISION FOR A HETEROGENEOUS MULTI-CHAIN FRAMEWORK DRAFT 1	2
1.1	概要	2
1.2	1. 序章	2
1.3	2. イントロダクション	3
1.4	まとめ	8
2	Polkadot に参加する	9
2.1	4.1. バリデーター	10
2.2	4.2. ノミネーター	11
2.3	4.3. コレイター	11
2.4	4.4. フィッシャーマン	12
2.5	5.1. コンセンサス	13
2.6	5.2. 掛け金を証明する	13
2.7	5.3. Parachain とコレイター	15
2.8	5.4. インターチェーンコミュニケーション	16
2.9	5.5. Polkadot と Ethereum	18
2.10	5.5.1. Polkadot から Ethereum	18
2.11	5.5.2. Ethereum から Polkadot	19
2.12	5.6. Polkadot と Bitcoin	20
3	6. プロトコルの詳細	21
3.1	6.1. リレーチェーンオペレーション	21
3.2	6.3. パラチェーンレジストリ	25
3.3	6.4. リレーブロックのシーリング	27
4	7. プロトコルの実用性	43
4.1	7.1. チェーン間のトランザクション支払い	43
4.2	7.2. 追加のチェーン	43
4.3	8.1. 欠けている資料と未解決の質問	44
4.4	8.2. 謝辞	45
4.5	APPENDIX A. 機能コンポーネント	45
5	APPENDIX B. FREQUENTLY ASKED QUESTIONS	49

1 POLKADOT: VISION FOR A HETEROGENEOUS MULTI-CHAIN FRAMEWORK DRAFT 1

DR. GAVIN WOOD FOUNDER, ETHEREUM & PARITY GAVIN@PARITY.IO

翻訳者： - Sota Watanabe (Twitter @WatanabeSota) - Masaki Minamide (Twitter @raika_5179)

1.1 概要

現在のブロックチェーンアーキテクチャは拡張性やスケーラビリティに留まらず、様々な問題点を抱えている。私達はこの理由を、簡潔さと正当性という2つのコンセンサスアーキテクチャの重要な2つの要素が密接に絡むことに起因すると考えている。このペーパーでは、この2つの要素を合わせた混合のマルチチェーンのアーキテクチャを紹介する。これらを2つに区分するとし、最低限のセキュリティと移送性の制限化でも全ての機能性を保つことによって実現した、状態に応じたコアの拡張性の実用的な用途を解説する。スケーラビリティはこれらの2つの機能に対して、分割し統治せよのアプローチで対処されている。信頼されていないパブリックのノードのインセンティブ設計を通してスケーリングアウトするように設計されている。このアーキテクチャの混合性はトラストレスかつ、完全に分散化した“federation”の中で相互運用性を持ち、オープンないしクローズのネットワークがお互いにトラストフリーのアクセスを行うことができ、様々なタイプのコンセンサスシステムに対応することができる。私達は1つ以上のEthereumのような既存のネットワークとバックワードで適合性をもつ方法を実装する。そのようなシステムは世界で商用レベルのスケーラビリティとプライバシーの達成を実装できるシステムとしてベースレベルの要素として提供され则认为している。

1.2 1. 序章

このペーパーは現実的にさらなるブロックチェーンのパラダイムを作っていく過程で取りうる1つの方向性を示唆した技術的な”ビジョン”のサマリーである。また、ブロックチェーン技術の様々な観点での具体的な改善策を提供する開発システムを現時点で可能な限り詳細を述べる。

このペーパーは、公式な詳細仕様書であることを意図していない。また、包括的な最終デザインであるわけでもない。APIやバインディング、言語や使用

方法をカバーすることもしない。パラメーターは特定されているが変更が見込まれる極めて実験的なペーパーである。コミュニティのアイデアや批評によってメカニズムが追加されるかもしれないし、修正、削除されるかもしれない。実験的なエビデンスとプロトタイピングによって何ができて何ができないのかがわかるので、このペーパーの大部分が修正されることもありうる。このドキュメントには様々な部分をより良くするかもしれないアイデアを含むプロトコルの詳細が述べられている。その詳細は初期 Proof-of-Concept (実証実験) のベースになるものとして期待されている。最終的に完成する “version 1.0” は本プロジェクトの目的を達成するためのさらなるアイデアが反映され、修正されたプロトコルが基になるだろう。

1.2.1 1.1 歴史

- 09/10/2016: 0.1.0-proof1
- 20/10/2016: 0.1.0-proof2
- 01/11/2016: 0.1.0-proof3
- 10/11/2016: 0.1.0

1.3 2. イントロダクション

ブロックチェーンは “Internet of Things” (IoT)、ファイナンス、ガバナンス、アイデンティティマネジメント、分散ウェブ、アセットトラッキングなど様々なフィールドで実用的であることを証明してきた。しかし、技術の素晴らしさと誇張された話しの裏腹で、ブロックチェーンが実社会に多大な影響を与えているというわけではない。私達はこれは現在のテクノロジースタックの5つの鍵となる問題点があるからだと考えている。

スケーラビリティ：単一のトランザクションがシステム上で処理されるまでバンドウィズ、ストレージを含め全体でどれくらいリソースを費やしているか。そして、最大でどれくらいのトランザクションが処理できるか？

孤立性：同じフレームワーク下で複数の参加者、アプリケーションの様々なニーズにどれくらい対処することができるか？

開発可能性：ツールがどれくらいよく動くか？ API は開発者のニーズに答えられているか？ 教育用のツールは整っているか？ 正しいインテグレーションがあるか？

ガバナンス：ネットワークが柔軟に何度も進化し変更する余地が残されているか？ 包括的に意思決定ができる仕組みか？ 効率的なリーダーシップをもたらす正当性と透明性がある分散システムか？

適応性: 技術が必要としているニーズにあっているか？ 現実のアプリケーションとのギャップを埋めるために”ミドルウェア”が必要か？

現時点では、私達は最初の2つの問題に着手するつもりであるが、Polkadotのフレームワークがこれらの問題に多大な改善をもたらすことができると信じている。Parity Ethereumのような実用的なブロックチェーンの実装は、性能の良いハードウェアを用いると秒回3000トランザクションを超える処理が可能である。しかし、現実のブロックチェーンネットワークでは秒間30トランザクションに制限されている。この制限は主に同期を取るコンセンサスメカニズムが、安全性のために時間を要するように設計されているから存在している。これは根底にあるコンセンサスアーキテクチャーによるものである。state transition mechanismはトランザクションを収集し処理する過程で様々な正当性や歴史に同意をとり、かつ「同期」するメカニズムである。

このシステムはproof-of-work (PoW) システムで可動しているBitcoinやEthereumや、proof-of-stake (PoS) で可動しているNXTやBitsharesにも同じく適応できる。(注、EthereumはPoSに移行) これらはどれも同じハンディキャップを抱えている。この問題を解決できればブロックチェーンが更に良いものになることは間違いないが、これらの2つのメカニズムを1つのプロトコルで扱うには、リスクやスケーラビリティ能力、プライバシー要求が異なる様々な全く別の主体やアプリケーションと一緒に扱わなければならない。1つのサイズではまったく適さないのである。Too often it is the case that in a desire for broad appeal, a network adopts a degree of conservatism which results in a lowest-common-denominator optimally serving few and ultimately leading to a failing in the ability to innovate, perform and adapt, sometimes dramatically so.

Factomのようないくつかのシステムはstate transitionメカニズムを取り入れていない。しかし、私達が望む用途の多くは共通したstate-machineを持つことによるtransition stateを可能にすることを求められている。

従って、スケールする分散コンピュータシステムを開発する合理的な打ち手はコンセンサスアーキテクチャーをstate-transitionメカニズムから切り離すことであることは明らかであるかのように見える。そして、驚くことではないかもしれないが、これがPolkadotがスケーリングソリューションである所以なのだ。

1.3.1 2.1 プロトコル、実装、ネットワーク

Bitcoin、Ethereumと同じように、Polkadotはネットワークプロトコルとそのプロトコルで動くパブリックネットワークであると言及される。PolkadotはCreative Commons licenseでコードはFLOSS licenseの下、無料でオープン

ンなプロジェクトを意図して作成されている。オープンソースで開発されているこのプロジェクトは誰であれコントリビューションを行うことができる。RFCs のシステムは、Python Enhancement Proposals と同じようにプロトコルの変更やアップデートにあたり公開された形で共同開発できるように設計されている。API を含む Polkadot プロトコルの私達の最初の実装は Parity Polkadot Platform として知られることとなるだろう。Parity の他のブロックチェーンの実装と同じように PPP はパブリックブロックチェーンやプライベート/コンソーシアムブロックチェーンに限らず汎用目的ブロックチェーン技術として開発されており、開発はイギリス政府を含むいくつかの団体からの補助金を基に行われている。このペーパーは言うまでもなくパブリックネットワーク下で Polkadot を描写している。パブリックネットワークで私達が思い描く機能は他のネットワーク（例：パブリック or/and プライベート）の上位互換である。さらに、この文脈で Polkadot の全容がさらにはっきり明記され議論することもできるだろう。これはつまり、読者があるメカニズムのどれが Polkadot に関連しているか、いつパブリックでない環境にデプロイされたかなどを注意しなければならないということである。

1.3.2 2.2. 先行研究

コンセンサスを state-transition から切り離す方法というのは正式にはないが、少なくとも 2 年間は提唱され続けている。この方法の提唱者である Max Kaye は Ethereum のかなり初期のメンバーでもあった。2014 年 6 月にまでさかのぼりその次の年に公開された Chain fibers として知られる複雑なスケラブルソリューションは単体の Relay-chain と透明性の高いインターチェーン処理メカニズムを提供する混合の複数のチェーンチェーンを実装した。Decoherence was paid for through transaction latency—transactions requiring the coordination of disparate portions of the system would take longer to process. Polkadot はその大部分のデザインと設計は異なるものの、アーキテクチャーの多くは参考にしている。Polkadot と比べうるシステムは実際のところ存在していないけれど、他のいくつかのシステムで結局は些細な部分であるが類似点が提案されているということもある。それらの提案をブレイクダウンするとグローバルに一貫性のある state machine を細かくしたもののである。

1.3.2.1 2.2.1. global State のないシステム Factom は適切な検証なしの正当さとデータの同期を許すことによる効率さを実証したシステムである。global state とスケラリングに紐づく difficulties を避けたことにより、スケラブルなブロックチェーンだとされている。しかし、以前にお話したとおり、これが解決した問題は限定的であり制約がある。

Tangle はコンセンサスシステムに対する斬新なアプローチである。Rather than arranging transactions into blocks and forming consensus over a strictly linked list to give a globally canonical ordering of state-changes, it largely abandons the idea of a heavily structured ordering and instead pushes for a directed acyclic graph of dependent transactions with later items helping canonicalise earlier items through explicit referencing. For arbitrary state-changes, this dependency graph would quickly become intractable, however for the much simpler UTXO model² this becomes quite reasonable. Because the system is only loosely coherent and transactions are generally independent of each other, a large amount of global parallelism becomes quite natural.

Using the UTXO model does have the effect of limiting Tangle to a purely value-transfer “currency” system rather than anything more general or extensible. Furthermore without the hard global coherency, interaction with other systems—which tend to need an absolute degree knowledge over the system state—becomes impractical.

1.3.2.2 2.2.2. Heterogeneous Chain Systems サイドチェーンは Bitcoin プロトコルで提案された追加仕様であり、メインのビットコインと追加となるサイドチェーン間でトラストレスなやりとりを可能にするプロトコルである。サイドチェーン間での”濃い”やりとりは想定されていない。つまり、やりとりは制限されており両者のアセットの交換点 (2 way peg 点) で管理者が存在する。The end vision is for a framework where the Bitcoin currency could be provided with additional, if peripheral, functionality through pegging it onto some other chains with more exotic state transition systems than the Bitcoin protocol allows. この意味で、サイドチェーンはスケーラビリティというよりはむしろ拡張性をもたらすものである。

もちろん、サイドチェーンの正当性に関しては根本的に用意されたものはない。一つのチェーン (例: Bitcoin) 上のトークンはマイナーに正当な正しいトランザクションを検証してもらうというサイドチェーンの能力によってのみ安全性が担保されている。Bitcoin のネットワークの安全性を他のブロックチェーンに移植することは簡単にできることではない。さらに、Bitcoin のマージマイニングを行うためのプロトコルはこのペーパーの対象外である。

Cosmos は Nakamoto PoW コンセンサスメソッドを Jae Kwon の Tendermint アルゴリズムに変えたマルチチェーンシステムである。本質的には、Tendermint の個々のインスタンスを使って Zone で運営されるマルチチェーンが master hub chain を介しトラストレスなコミュニケーションを可能にする。このインターチェーンコミュニケーションは任意の情報というよりはデジタル・ア

セット（またの名をトークン）の移動に制限されている。しかし、そのようなインターチェーンコミュニケーションは data の受け渡しも可能であるといえれば可能である。

Validator sets for the zoned chains, and in particular the means of incentivising them, are, like side-chains, left as an unsolved problem. 一般的な予測では、各々の Zone チェーン自体でトークンを保有しており、その上昇価値がバリデーターに還元される。未だに、まだ初期のデザインなので、一貫性のある詳細を確認することはできないが、Zone と Hub 間のゆるい一貫性は Zone チェーンのパラメーターに柔軟性を出すだろうと考えられる。

1.3.2.3 2.2.3. Casper 2つの内1つがもう一方を不要にするという見解があるが、Polkadot と Casper に関する包括的な比較はまだ存在しない。Casper はどちらのフォークしたチェーンが正当になるのかについて PoS アルゴリズムによって参加者が賭けをした記録に基づいている。Substantial consideration was given to ensuring that it be robust to network forks, even when prolonged, and have some additional degree of scalability on top of the basic Ethereum model. なので、Casper は本質的に Polkadot や派生系よりも複雑なモデルであると言えるかもしれない。いかに Casper が将来発展するのか最終的にどのような形でデプロイされるのかは依然として不明である。

Casper も Polkadot も両者とも興味深い新しいプロトコルを提案している。そして、いくつかの点で Ethereum の究極の目的と開発方向が食い違っている。Casper は Ethereum Foundation が手動するプロジェクトで、元々は完全なスケーリングを意図しないプロトコルの PoS への仕様変更であった。重要なことに、それは Ethereum のすべてのクライアントにアップデートを要求するハードフォークであった。なので、開発は強い連携が必要であった分散プロジェクトを継承したものよりも本質的にさらに複雑になった。

Polkadot はいくつかの点で異なる。まず、第一に Polkadot は完全に拡張性が高くスケラブルなブロックチェーン開発である。It is built to be a largely future-proof harness able to assimilate new blockchain technology as it becomes available without over-complicated decentralised coordination or hard forks. 私達は秘匿化されたコンソーシアムチェーンの運用運用や Ethereum では想定できないようなブロック生成時間の短いチェーンなどすでにいくつかのユースケースを想定している。最後に、Polkadot と Ethereum 間の結合は極めてゆるい。2つのネットワーク間でトラストレスなやり取りをすることができる。

一言で言うと、Casper/Ethereum 2.0 と Polkadot はかすかに類似点を持つけれども、目的地が明らかに異なると考えている。そして、近い将来、競合するというよりはむしろ、相互に恩恵のある形で併存することになるだろう。

1.4 まとめ

Polkadot はスケーリングを見込んだ異種混合のマルチチェーンである。つまり、一般的なアプリケーションに特化したいままの単一ブロックチェーンの実装とはことなり、Polkadot それ自体はアプリケーションを継承する構造をとっていない。むしろ、Polkadot は “relay-chain” という基盤をもっており、その上に沢山の検証性を持つグローバルで一貫性のあるダイナミックなデータ構造が繋がる。我々はそれらのデータ構造を “parallelised” チェーンもしくは parachains と呼んでいる。言い換えると、Polkadot は 2 つの重要な点を除けば自立したチェーンのセットと同じようなものだと考えられる。(例: Ethereum, Ethereum Classic, Namecoin と Bitcoin のおセット)

- プールされたセキュリティ
- トラストフリーインターチェーントランザクション

これらの点は、私達が Polkadot がスケールすることができると考えている理由である。In principle, a problem to be deployed on Polkadot may be substantially parallelised—scaled out—over a large number of parachains. それぞれのパラチェーン全ての機能が Polkadot ネットワークの異なるセグメントで同時処理されるため、システムにはスケールする余地がある。Polkadot はインフラストラクチャーのコア部分をミドルウェアレベルでは複雑な仕様を扱えるようにしながら提供している。この背景には、開発リスクを減らし短時間で効率的な開発ができるように、そして安全性と堅牢性を備えることができるように設計するという重要な意思決定がある。

1.4.1 3.1. Polkadot の哲学

Polkadot はその上に、萌芽期のアイデアから熟練したデザインまで対応する次の波となるコンセンサスシステムを実装する極めて堅牢な基盤を提供するべきだ。安全性、独自性、相互通信性に関して強い保証を提供することで、Polkadot は parachain 自体に拡張範囲を選択させている。当然のことながら、私達は様々なブロックチェーンの実験が実用的な構成要素の開発の手立てになると考えている。

私達は Bitcoin or Z-cash のような保守的で、高価値が乗っているチェーンが価値が乗っていない “theme-chains” (マーケティング目的や遊びで作ったもの) と 0 ないしほぼ 0 料金のテストネットと共存すると考えている。また、完全に暗号化された “暗い” コンソーシアムチェーンが、機能性に長けオープンである Ethereum のようなチェーンとですら繋がると考えている。成熟した Ethereum や型が定義されている Bitcoin のようなチェーンから計算難易度の高い計算をアウトソースされた WASM チェーンといった実験的な VM ベースですら共存するだろう。

チェーンのアップグレードを管理するために、Polkadot はできるだけ既存のシステムと Yellow paper でいう Council と類似した 2 院制に基づいた固有のガバナンス構造をサポートしている。

絶対的な権限としてトークンホルダーは” 一般投票” をコントロールするを持つ。ユーザーの開発ニーズだけではなく、開発者のニーズを満たすために、私達は、バリデーターによる” ユーザー” の議会と開発者とエコシステムの参加者によって成り立つ” 技術的な” 議会” の 2 つが良い方向へ導いてくれることを期待している。トークンホルダーの核は絶対の正当性を保持し、多数の意見を增強したり、パラメーターで示したり、取替えたり、分解したりすることだ。Twain の言葉を借りれば、“政府とおむつはよく取り替えなければいけない。どちらも同じ理由で。” である。

一方で、大規模なコンセンサスメカニズムを調整するためにパラメータを再構成するのは、ささいなことである。more qualitative changes such as replacement and augmentation would likely need to be either non-automated “soft-decrees” (e.g. through the canonicalisation of a block number and the hash of a document formally specifying the new protocol) or necessitate the core consensus mechanism to contain a sufficiently rich language to describe any aspect of itself which may need to change. The latter is an eventual aim, however, the former more likely to be chosen in order to facilitate a reasonable development timeline. Polkadot の全てのデザイン決定における主な信条とルールは

最小限であること: Polkadot はできるだけ少ない機能で実装する

シンプル: 一般的にミドルウェア、Parachain もしくは後の実装に負荷をかけるベースプロトコルの複雑さを最小限に抑える

一般的であること: 不要な実装を避ける。制約や限界を Parachain に設ける。Polkadot はどのモデルが一番堅牢かを最適化するコンセンサスシステム開発の基盤であるべきである。

堅牢であること: Polkadot は根本的に安定したベースレイヤーであるべきである。経済的な側面に加え、高いインセンティブをもつ攻撃の可能性を最小化する分散システムであることを意味する。

である。

2 Polkadot に参加する

Polkadot には 4 つの基本的な役割がある。すなわち、コレイター、フィッシャーマン、ノミネーターとバリデーターである。Polkadot の 1 つの取りう

る実装では最後の役割は基本的なバリデーターと有用性の保証人という 2 つの役割に分けられる。これは後で議論されることになっている。

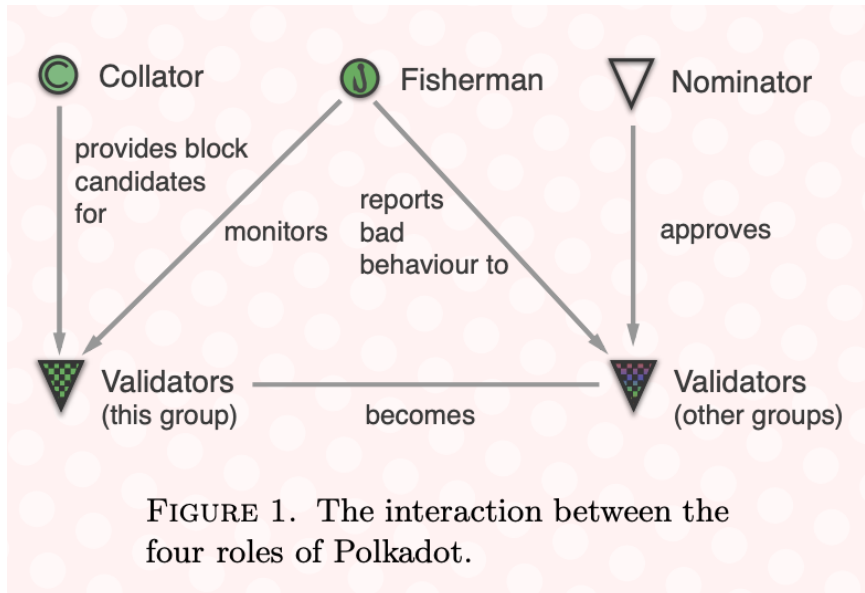


図 1: Figure1

2.1 4.1. バリデーター

バリデーターは Polkadot ネットワークを管理し、新しいブロックを認証する。バリデーターの役割は十分に高額な掛け金がデポジットされていることに依存している。これは、他の掛け金が賭けられている他のグループから 1 つ以上のバリデーターを選出し同じように振る舞わせることもできる。なので、バリデーターの掛け金の一部はバリデーターに管理される必要はなく、むしろノミネーターによって管理されていると言える。バリデーターは高い有用性と帯域が必要なクライアントを relay-chain で走らせなければならない。それぞれのブロックでノードはノミネートされた parachain で新しいブロックを検証する準備をしなければならない。このプロセスには候補となるブロックの受け取り検証、再発行が含まれている。ノミネーションは決定的であるが、現実的には事前に予測不可能である。バリデーターが全ての parachain のデータベースを全て同期するとは合理的に考えて考えられないので、バリデーターは提案された新しい parachain のブロックを確定する役割をコレイターとして知られる第三者に委託する。

全ての新しい parachain ブロックが予定されたバリデーターのサブグループによって適切に検証されたら、バリデーターは relay-chain のブロック自体を

検証する。この作業にはトランザクション State 文字列のアップデート（本質的には paracahin のアウトプット文字列を他の paracahin のインプット文字列にデータを移すこと。）、検証された relaycahin のトランザクションセットのトランザクションを処理すること、最後の parachain の最終変更が含まれるファイナルブロックの承認が含まれている。

バリデーターは私達が選んだコンセンサスアルゴリズムのルール下で彼らの責任を満たさない行動を起こした場合、罰せられる。意図的ではない障害でも、バリデーターの報酬は差し控えられる。繰り返される機能停止は結果的にセキュリティボンドの減少を招く（バーンを通して）。2重支払いや不正ブロックの共謀などの悪意ある攻撃によって全ての掛け金を失うことになるかもしれない。いくつかの観点で、バリデーターは現在の PoW チェーンにおけるマイニングプールに似ている。

2.2 4.2. ノミネーター

ノミネーターはバリデーターのセキュリティボンドに貢献する stake-holding party である。リスクキャピタルを持ち、そのことによって彼らがネットワークをメンテナンスしている特定のバリデーター（もしくはグループ）を信頼していることを表明すること以外に追加での役割はない。彼らは掛け金の増加、減少に応じて対価を受け取る。次で説明するコレイターと一緒に、ノミネーターは PoW ネットワークにおけるマイナーと似ている。

2.3 4.3. コレイター

トランザクションコレイター（略してコレイター）は、バリデーターが正当な paracahin ブロックを生成するのをサポートするグループである。彼らは、特定の parachain の “full-node” を持つ。これは、現在の PoW ブロックチェーンにてマイナーがしているのと同じように、新しいブロックを監視しトランザクションを実行する為に必要な情報を保持している。普通の状態では、まだ承認されていないブロックを生成するために、トランザクションを照合し実行する。そして、ゼロ知識証明と共にブロックを parachain のブロックを提案する責任をもっているバリデーターに伝播する。

コレイター、ノミネーター、バリデーターの正確な関係は時間とともに変更される可能性が高い。初期は、少数の paracahin とトランザクションが想定されるのでコレイターはバリデーターと密接に働くことを想定している。イニシャルのクライアント実装は parachain のコレイターノードが無条件に正当だとされている paracahin のブロックを relaycahin のバリデーターに提供するために RPC を含んでいる。同期されたバージョンの全てのパラチェー

ンの保管コストが増加するので追加で、そのコストを分散化する経済的インセンティブ設計のあるグループが活動するインフラを作ることが見込まれる。

最終的には、ほぼすべてのトランザクションフィーを回収しようと競争するコレイタープールの存在を期待している。そのようなコレイターは報酬のシェアを目的として特定のバリデーターと一定期間契約を結ぶようになるかもしれない。

代用となる”フリーランス”的なコレイターは may simply create a market offering valid parachain blocks in return for a competitive share of the reward payable immediately. シンプルに、分散化されたノミネーターのプールは掛け金を入れている複数の参加者にバリデーターの仕事をシェアしコーディネートする。これはプールにおけるオープンな参加モデルはさらに分散化したシステムの構築をもたらす。

2.4 4.4. フィッシャーマン

他の2つの役割とは異なり、フィッシャーマンはブロックの承認プロセスに直接関わってはいない。むしろ、報酬によってモチベートされた “bounty hunters” として独立している。正確にはフィッシャーマンの存在によってめったに不正行為が起きないことが想定されている。そして、もし不正行為が起これるのであれば悪意ある攻撃というよりもシークレットキーのセキュリティに注意が足りなかったときである。The name comes from the expected frequency of reward, the minimal requirements to take part and the eventual reward size.

フィッシャーマンは少なくとも1つの掛け金がされている参加者が不正な行為をしたことを時間内に証明することで報酬が得られる。paracahin の場合は、不正行為は同じ鍵で2つの異なるブロックに署名をしたり、不正なブロックを承認するのを手伝ったりすることである。規格外の報酬やセッションのシークレットキーを悪用するのを防ぐために、単一のバリデーターの悪意ある署名メッセージを提供することのベース報酬は最小限に抑えられている。この報酬は他のバリデーターも同様に悪意ある攻撃を仕掛けようとしていた場合、漸近的に増加する。最低でも2/3のバリデーターが善意のある行動をしているというベースセキュリティでは、漸近率は66%に設定されている。

フィッシャーマンはいくつかの点で必要とされているリソースが相対的に少量で安定性と帯域がそれほど必要ではない現在のブロックチェーンシステムにおける”full nodes” に似ている。フィッシャーマンは少量の掛け金を支払うという点で異なる。この掛け金はバリデーターの時間とコンピューテーションリソースを奪うという点でシビリアタックを防ぐ効果がある。掛け金はす

ぐに引き出すことができる。これはおそらく数ドル程度で悪意あるバリデーターの攻撃を防ぐことによって大量の報酬を得ることができる。

このセクションではシステム全体の設計像を簡潔に説明する。システムの詳細な説明は章を追って解説する。

2.5 5.1. コンセンサス

relay-chain では、Polkadot は正当なブロックが同期性のある Byzantine fault-tolerant (BFT) アルゴリズムを通して合意される。このアルゴリズムは Tendermint [11] によってインスパイアされたものである。そして、副次的に HoneyBadgerBFT [14] に似通っている。後者は任意の不完全性のあるネットワークインフラで正常に動作する権威者もしくはバリデーターがいれば効率的で fault-tolerant のコンセンサスを提供する。

proof-of-authority (PoA) スタイルのネットワークでは、これだけで十分である。しかし、Polkadot は信頼される権威者や 3rd パーティーが存在しない完全に Open でパブリックな環境ネットワークとしてデプロイが可能であるように設計されている。なので私達にはバリデーターを決定し、正直に動くためにインセンティブ設計する必要がある。なので、PoS ベースの選考基準を設ける。

2.6 5.2. 掛け金を証明する

私達はネットワークに特定のアカウントがいくらの”掛け金”を持っているのかを測る方法があることを想定している。既存のシステムと比較しやすいように計測する単位を”トークン”とする。この言葉はいくつかの理由で理想的ではない。1 つはアカウントに紐付いている値がスカラー値であるとは限らないこと。もう 1 つは個別のトークンに特有性がないからである。

私達は、頻度は高くないが（最大でも 1 日に 1 回、もしかすると 4 半期に 1 回ほど）Nominated Proof-of-Stake (NPoS) によってバリデーターは選出されることを想定している。Incentivisation can happen through a pro-rata allocation of funds coming from a token base expansion (up to 100% per year, though more likely around 10%) together with any transaction fees collected. マネタリーベースの膨張は主にインフレーションをもたらすけれども、全てのトークン保有者が参加権をフェアに持つので、時とともに価値がへるという心配をする必要がない。そのことでコンセンサスメカニズムにおけるトークンホルダーの役割を喜んでこなすようになる。トークンの特定の割合がステーキングプロセスのターゲットとなる。効率的なトークンベ-

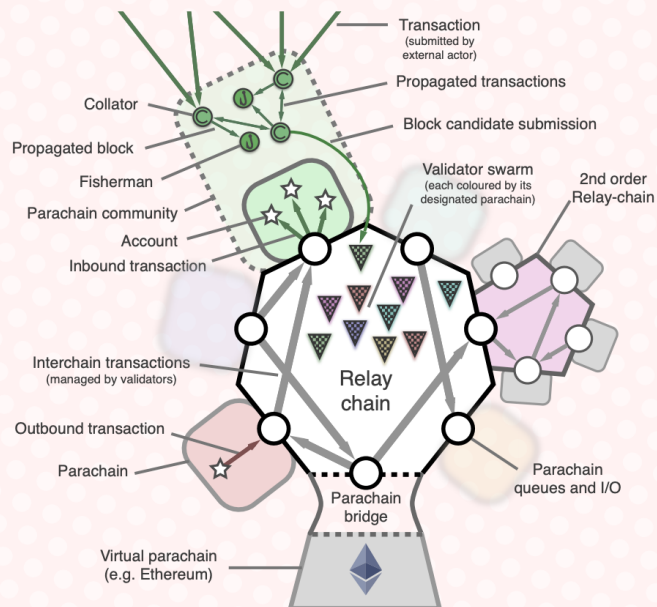


FIGURE 2. A summary schematic of the Polkadot system. This shows collators collecting and propagating user-transactions, as well as propagating block candidates to fishermen and validators. It also shows how an account can post a transaction which is carried out of its parachain, via the relay-chain and on into another parachain where it can be interpreted as a transaction to an account there.

☒ 2: Summary

スの増加 (token base expansion) はマーケットベースのメカニズムでこのターゲット値に近づくように調整される。

バリデーターは掛け金によって密接につながっている。バリデーターが掛け金を引き出すことができるのは、バリデーターの責務が終わり時間が経ってからである。(3ヶ月くらい) この長い期間が存在するのは、チェーンのチェックポイントまでバリデーターが将来不正を行った場合罰するためである。報酬額を削減したり、意図的にネットワークを悪化させたり、他のバリデーターに掛け金を渡したりした場合、罰を受けることになる。例えば、バリデーターがフォークした両方のチェーンを承認しようとした場合 (しばしばショートレンジアタックと呼ばれる)、後々それが検知され罰せられる。

Long-range “nothing-at-stake” attacks⁴ are circumvented through a simple “checkpoint” latch which prevents a dangerous chain-reorganisation of more than a particular chain-depth. 新しく同期してきたクライアントが間違ったチェーンに騙さないことを確認するために、バリデーターの掛け金整理のタイミングと同じく定期でハードフォークが行われる。これによって最新のチェックポイントブロックハッシュがクライアントに入る。これは今後、“finite chain length” やジェネシスブロックを定期的にリセットすることによってうまくいくようになる。

2.7 5.3. Parachain とコレイター

各 parachain は relay-chain に近いセキュリティ強度を持つ。parachain のヘッダーは relay-chain のブロックに格納されており、再編成や 2 重支払いがないようにしている。これは Bitcoin でいうサイドチェーンとマージマイニングと似たセキュリティ保証である。Polkadot ではそれに加え、parachain の state transaction が正当であるという強い保証を提供する。これは、バリデーターが暗号的にランダムにサブセットに配属されることを通して行われる。parachain ごとにサブセットがあるかもしれない、block ごとにサブセットはことなるかもしれない。この設定は parachain のブロックタイムが relay-chain のブロックタイムと少なくとも同じくらい長いことを示している。この分割の詳しい方法はこの論文の対象外である。しかし、RanDAO に似た commit-reveal フレームワーク、もしくは暗号的に安全なハッシュ下の各 parachain ブロックに結合したデータに立脚している可能性が高い。

そのようなバリデーターのサブセットは正当だと保証された parachain のブロック候補を提供することを要求されている。正当性というのは 2 つの重要なポイントを含む。1 つ目は全ての state transitions が正確に実行され、全ての参照される外部データが最終的に正当であるということ。2 つ目は、that any data which is extrinsic to its candidate, such as those external transactions,

has sufficiently high availability so that participants are able to download it and execute the block manually.⁵ Validators may provide only a “null” block containing no external “transactions” data, but may run the risk of getting a reduced reward if they do. They work alongside a parachain gossip protocol with collators—individuals who collate transactions into blocks and provide a noninteractive, zero-knowledge proof that the block constitutes a valid child of its parent (and taking any transaction fees for their trouble).

parachain のプロトコルにチェーン独自のスパム防止方法を搭載する余地を残している。relay-chain にある “compute-resource metering” もしくは、“transaction fee” という根本的な概念は存在しない。relaychain プロトコルによってこれらが強制されることもない。(しかし、堅牢なメカニズムの用意されていない parachain をステークホルダーが採用することは起こりえないだろう) これは Ethereum のようなチェーンと明確に異っている。(例：シンプルな fee モデルを持つ Bitcoin のようなチェーンもしくは、スパム防止モデルは提唱されていないがそれ以外)

Polkadot の relay-chain それ自体は Etheruem のような accounts、state チェーンとして存在する可能性が高い。もしかすると、EVM の派生系であるかもしれない。relay-cahin のノードはかなりの処理能力、トランザクションスループットが要求されるので、トランザクションスループットは高いトランザクション手数料とブロックサイズリミットによって最小化されるだろう。

2.8 5.4. インターチェーンコミュニケーション

Polkadot の重要な最後の要素は、インターチェーンコミュニケーションである。paracahin 間では幾分かのインフォメーションチャネルが存在するので、Polkadot はスケーラブルなマルチチェーンであると私達は考えている。Polkadot の場合、コミュニケーションはできるだけシンプルに設計している。paracahin で処理されるトランザクションは (Chain のロジックによっては) 2 つ目の parachain もしくは、relay-cahin に発送することができる。商用的なブロックチェーン上の外部トランザクションのように、トランザクションは完全に非同期であり情報をもとの出处に返し継承することはできない。

実装の複雑性、リスク、将来の parachain アーキテクチャーの制限を最小化する為に、これらのインターチェントランザクションは一般的な外部トランザクションと区別することができない。トランザクションは paracahin を認識することができるオリジナルの要素と任意のサイズのアドレスを持つ。Bitcoin や Ethereum といった現在の一般的なシステムとは異なり、インターチェントランザクションでは料金が紐づく”決済”はできない。そのような決済は発生源と目的地の parachain 間のネゴシエーションロジックを通して管理

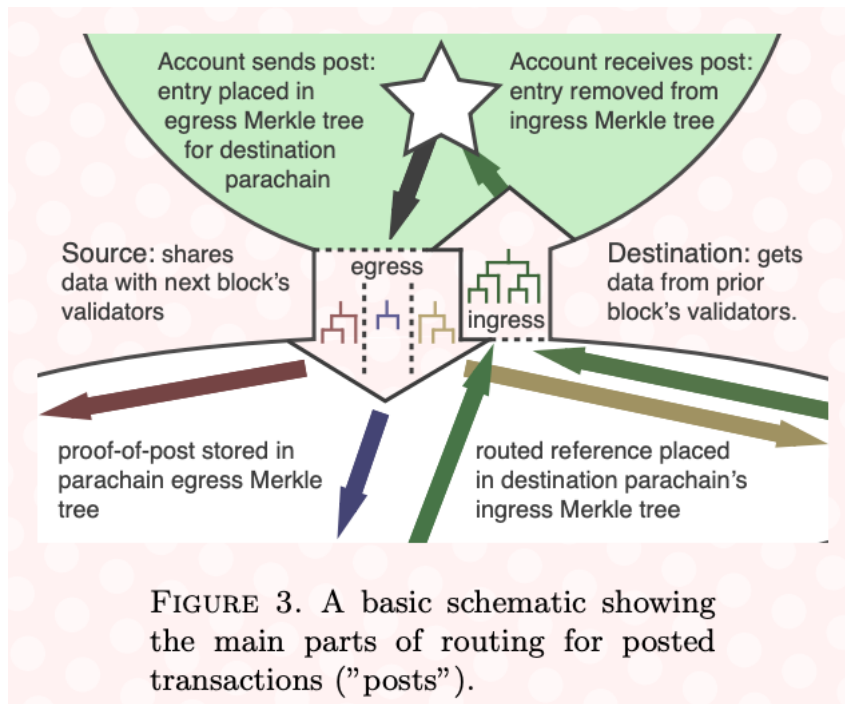


図 3: transaction

されなければならない。Ethereum' s Serenity で提案されているシステムはそのようなクロスチェーン決済を管理する簡単な方法である。

インターチェントランザクションは正確性を保証する Markle tree ベースの単純な行列メカニズムを使うことによって解決される。1つの paracahin のアウトプット列を目的地の paracahin のインプット列に移動させるのは relay-cahin のメンテナーの仕事である。その送られたトランザクションは relay-cahin のトランザクションそれ自体ではなく、relay-chain によって参照される。他の paracahin のトランザクションから paracahin にスパムが送られるのを防ぐために、目的地のインプット列は一番前のブロック列の時よりも大きすぎないようにする必要がある。もしそのインプット列がブロック処理後、大きすぎた場合、それは” 仕込まれた” と考えられ、限界値以下に減らさない限り、続くブロックにおいてトランザクションを処理することができなくなる。それらの列は relay-cahin によって監督され、互いの paracahin の状態を監視し合うことができる。なので、不正の目的地にトランザクションが送信された場合、一瞬で報告され目論見は失敗する。(リターンパスが存在しないので、2次トランザクションがこの理由で失敗した場合、オリジナル caller に報告することができない。他のリカバリー方法が実行される必要がある。)

2.9 5.5. Polkadot と Ethereum

Ethereum のチューリング完全性により、お互いにインターオペラビリティを持つ可能性が豊富にあると考えている。少なくとも、セキュリティを共有することはできると思う。簡単に言うと、私達は Polkadot からのトランザクションはバリデーターによって署名され、Ethereum 上で運用することができ、送信先のコントラクトを起動できるということである。In the other direction, we foresee the usage of specially formatted logs (events) coming from a “break-out contract” to allow a swift verification that a particular message should be forwarded.

2.10 5.5.1. Polkadot から Ethereum

承認投票メカニズムによって決定されたステークホルダーから形成されたバリデータによる BFT コンセンサスメカニズムを通じて、稀にしか変化せずそこまで多くないバリデータで安全なコンセンサスを得ることができる。合計 144 のバリデータ、4 秒のブロックタイム、900 ブロックのファイナリティ（二重投票などの悪意のある行動、処罰、修復）のシステムでは、ブロッは妥当であると考えられる。97 個の署名 (twothirds of 144 plus one) と 60 分の検証期間（デポジットとチャレンジなし）で問題がないことが証明されました。Ethereum は 144 人の署名者で維持される” break in contract”（侵入契約）をホストすることができる。EVM で ECDSA の回復にかかるガスはわずか 3000 ガスであるため、検証は（全会一致でなく）多くのバリデータによって行われるので、Ethereum の基本コストは Polkadot ネットワークからくるのは 300,000 ガス以下であると正しく検証された。-5.5M での総ブロックガスの制限はわずか 6% バリデータの数を増やすと（数十のチェーンを処理するのに必要となるように）必然的にこのコストが増加するが、技術の成熟とインフラの向上に伴い、Ethereum のトランザクションの帯域幅は徐々に拡大すると予想される。全てのバリデータが関与する必要があるわけではないという事実（例えば、そのようなタスクのために最もステークしたバリデータだけが要求される可能性がある）とともに、このメカニズムの限界は合理的に伸びている。このようなバリデータを毎日ローテーションすることを想定すると（これはかなり保守的で毎週または毎月さえも許容可能）この Ethereum 転送ブリッジは 1 日あたり約 54 万ガス、現在のガスコストでは 45 ドル。ブリッジを介して単独で転送された基本的なトランザクションはやく 0.11 ドルかかる。コントラクト計算にはもっとコストがかかる。バッファリングしたりトランザクションをまとめることで、侵入承認コストを簡単に共有できトランザクションあたりのコストを大幅に削減できる。20 のトランザクションが要求された場合、基本トランザクションの転送コストは約 0.01 ドルになる。このマルチシグネチャコントラクトモデルに代わる、興味深くて

安価な代替手段の一つは multi-lateral ownership semantic を実現するために閾値シグニチャを使用すること。ECDSA の閾値署名方式は計算量が大きくなるが Schnorr 署名などの他の方式のものよりは合理的である。Ethereum は今後の Metropolis ハードフォークでこのようなスキームを安価に使用できるようにするプリミティブを導入する予定である。そのような手段が利用できれば、Polkadot トランザクションを Ethereum ネットワークに転送するためのガスコストは大幅に削減され、オーバーヘッドはほぼゼロになり、署名を検証しトランザクションを実行する基本的なコストよりは高い。このモデルでは Polkadot のバリデータノードはメッセージにサインする以外にほとんど何もする必要がない。Polkadot トランザクションを Ethereum ネットワークに転送するためのガスコストは署名を検証し、基礎となるトランザクションを実行するための基本コストと比べて大幅に削減され、オーバーヘッドはほぼゼロになる。このモデルでは Polkadot のバリデータノードは、署名メッセージ以外にほとんど何もする必要がない。トランザクションを Ethereum ネットワークに実際にルーティングするためには、バリデータ自体も Ethereum ネットワーク上に存在するか、あるいはもっと可能性が高いのはメッセージをネットワークに転送する最初のアクターに少額の報奨金が提供されるかのいずれかを想定する。

2.11 5.5.2. Ethereum から Polkadot

Ethereum から Polkadot にトランザクションを転送するには、単純にログの概念を使用する。Ethereum のコントラクトでは、Polkadot の特定のパラチェーンにトランザクションをディスパッチする場合、特別な” break-out contract” を呼び出すだけで済む。” break-out contract” は必要とされる可能性のある全ての支払いを受け取り、マール証明と対応するブロックのヘッダが有効かつ正規であるというアサーションによってその存在が証明されるようにロギング命令を発行する。あとの二つの条件のうち、おそらく最も簡単に証明できるのは有効性である。原則として、各 Polkadot ノードに必要なのは、標準 Ethereum ノードの完全に同期されたインスタンスを実行するための証明（すなわち指定されたバリデータノード）だけである。残念ながらこれはこれ自体がかなり強い依存関係である。より軽量な方法は、ブロック内のトランザクションを適切に実行し、ログ（ブロック受信に含まれる）が有効であることを確認するために必要な Ethereum の state テストの一部のみを提供することによって、ヘッダが正しく評価されたという簡単な証明を使用すること。このような” SPV-like” 6 proofs はまだかなりの量の情報が必要である。しかし都合の良いことにこうしたヘッダは一般的に全てを必要としない。すなわちもし他の第三者（fisher など）がヘッダを無効であることを証明した場合（具体的には state root または receipt root が正しくない）polkadot 内のボンドシステムによって bond を所有する第三者は bond を失う危険を冒して

ヘッダを提出することができる。Ethereum のような non-finalize PoW ネットワークでは正規性を完全に証明するのは不可能である。これに対処するために、あらゆる種類のチェーン依存の原因と結果に依存しようとするアプリケーションはいくつかの” confirmations” を待つが、依存トランザクションがチェーン内のある特定の深さになるまで待つ。Ethereum ではこの深さは既知のネットワークの問題がない場合の最も価値の低いトランザクションの 1 ブロックから交換用に最初にリリースされた Frontier の場合の 1200 ブロックまで様々である。安定した” Homestead” ネットワークではこの数値はほとんどの交換で 120 ブロックになり、同様のパラメータをとることになる。したがって Polkadot 側 Ethereum インターフェースがいくつかの単純な機能を持つことを想像することができる。: Ethereum ネットワークから新しいヘッダーを受け入れ、POW を検証することができ、十分な深さ（対応するメッセージを Polkadot 内で転送する。）のヘッダのために Ethereum 側の” break out” コントラクトによって特定のログが出力されたという何らかの証明を受け入れることができ、最後に以前に受け入れられたがまだ実施されていないヘッダが無効な受信ルートを含むという証明を受け入れることができる。Ethereum ヘッダ、データ自体（および SPV の証明または有効性、妥当性の反論）を実際に Polkadot ネットワークに取り込むには、データ転送のインセンティブが必要である。これはヘッダが有効で有用なブロックを転送できる誰にでも支払われる支払い（Ethereum 側で徴収した料金から資金を得た）もののような単純な支払いである可能性がある。バリデータは何らかのプロトコル固有の手段によって、またはリレーチェーン上で維持されるコントラクトによって、フォークを管理することができるようにするために最後の数千ブロックに関する情報を保持することが求められる。

2.12 5.6. Polkadot と Bitcoin

Bitcoin とのインターオペラビリティは Polkadot にとって興味深い挑戦である。いわゆる” two way peg” は両方のネットワーク側にある有用なインフラストラクチャの一部になるだろう。しかし Bitcoin の限界のためにこのようなペグを安全に提供することは簡単ではない。Bitcoin から Polkadot への取引は原理的には Ethereum と同様のプロセスで行うことができる。Polkadot バリデータによって何らかの方法で制御される” break-out address” は転送されたトークン（データも一緒に送られる）を受け取ることができる。SPV の証明はインセンティブ化されたオラクルによって提供され、確認期間と共にトランザクションが” 二重消費された” ことを示す非正規ブロックを特定するための報奨金が与えられる。その後、” break-out address” で所有されるトークンは原則として後の disposal(分散?) のために同じバリデータによって制御される。しかし問題はデポジットをローディングバリデータセットから安全に制御する方法である。署名の組み合わせに基づいて任意の決定を行いこと

ができる Ethereum とは異なり、Bitcoin は実質的により限定されておりほとんどのクライアントは最大 3 人のパーティーによる複数の署名のトランザクションしか受け付けない。現在のプロトコルではこれを 36、あるいは最終的に数千に拡張することは不可能である。一つの選択肢は Bitcoin プロトコルを変更してそのような機能を可能にすることだが、Bitcoin の世界におけるハードフォークは最近の試みから判断すると調整が難しい。一つの可能性は閾値署名の使用である。これは単一の識別可能な公開鍵が複数のシークレット” parts” によって効果的に制御されることを可能にする暗号方式であり、そのいくつかまたは全ては有効な署名を作成するために利用されなければならない。残念なことに Bitcoin の ECDSA と互換性のある閾値署名は生成するには計算コストがかかり、多項式の複雑さを伴う。Schnorr 署名のような他のスキームははるかに低いコストを提供するが、それらが Bitcoin プロトコルに導入される時系列は不確実である。デポジットの最終的なセキュリティは多数のバリデータにかかっているため、他の一つの選択肢は” multi-signature key-holders” を閾値署名が実現可能になるように全体のバリデータのうちの強固に結合された部分集合のみに減らすことである（あるいは最悪の場合、Bitcoin のネイティブマルチシグネチャは可能だ。）これはもちろんバリデータが不正に行動した場合に賠償金として差し引かれる bonds の総額を減らすことができるが、これはゆうがな低下であり、単に二つのネットワーク間を安全に走ることができる資金量の上限を設定するだけである（あるいはバリデータからの攻撃が成功した場合の%損失について）。このように二つのネットワークの間に適度に安全な Bitcoin のインターオペラビリティ” virtual parachain” を置くことは非現実ではないと信じているが、不確実なスケジュールでの相当な努力が必要であり、そのネットワーク内のステークホルダーの協力を必要とする可能性が非常に高い。

3 6. プロトコルの詳細

本プロトコルは大きく 3 つのパートに分解することができる：コンセンサスメカニズム、パラチェーンインターフェイス、インターチェーン取引ルーティング。

3.1 6.1. リレーチェーンオペレーション

リレーチェーンはイーサリウムと似たように、ステイトがアドレスをアカウント情報（主に残高や取引回数）にマッピングしたステイトベースのチェーンになるだろう。アカウントをここに置くことには目的が一つある：システムで誰がどれだけのステイクを保持しているかを説明すること。そこには大きな違いはない、しかし：

- コントラクトはトランザクションによって配置することはできない。リレーチェーン上のアプリケーション機能を回避したいという欲求から、契約のパブリックデプロイのサポートをしない。
- 計算リソース（ガス）の使用量は計上されない；パブリック使用のための機能のみが直されるため、ガス計上はされなくなる。
- リストに挙げられたコントラクトが自動執行とネットワークメッセージアウトプットをすることを可能にする特殊な機能サポートされる。

リレーチェーンに VM があり、それが EVM をベースにしている場合、単純さを最大化するためにいくつかの変更が必要である。リレーチェーンにはコンセンサス、バリデーター、パラチェーンコントラクトを行うために、プラットフォーム専用のいくつかの組込みコントラクト（Ethereum の 1-4 アドレスのような）が存在している。

EVM ではない場合、WebAssembly [2]（wasm）バックエンドが最も可能性の高い代替手段である。この場合、全体構造は似ているが、埋め込みコントラクトの必要はない。これが EVM のために作られた未熟な言語ではなく汎用言語用である Wasm を使う理由である。

現在の Ethereum プロトコルからの全く異なる他のプロトコルの可能性は十分にある。例えば、Ethereum の Serenity のために提案されたように、同一ブロック内で競合しないトランザクションの並列実行を可能にする、簡易的なトランザクション受信フォーマットなど。

これは可能性は低いが、Serenity のような「純粋 (Pure)」なチェーンがチェーンの基礎的なプロトコルとしてではなく、Relay チェーンとしてデプロイされるかもしれない。これにより、特定のコントラクトがステイキングトークン残高などを管理することができる。現時点では、これは追加の複雑さと開発に不確実性を伴う価値がある程の十分なプロトコル簡素化を提供する可能性が低いと感じている。

コンセンサスメカニズム、バリデータセット、バリデーションメカニズムおよびパラチェーンを管理するために必要な機能のピースがいくつかもある。これらはモノリシックプロトコルの下で一緒に実装することが可能である。ただし、モジュール性を高めるために、これらをリレーチェーンの「コントラクト」として説明する。

これは、それらが（オブジェクト指向言語のような）オブジェクトであることを意味すると解釈される。リレーチェーンのコンセンサスメカニズムによって管理されているが、必ずしも EVM のような opcodes でプログラムとして定義されているわけでも、アカウントシステムを通じて個別にアドレス指定可能であるというわけでもない。

6.2. スтейキング コントラクト このコントラクトは以下のようにバリデータセットを管理する。

- どのアカウントが現在バリデータであるか (Validators) ;
- どのアカウントがすぐにバリデータになることができるか (Intentions) ;
- どのアカウントがバリデータにノミネートするためにステイクしているか (Stashes) ;
- スтейク量、許容ペイアウト率、アドレス、短期 (セッション) アイデンティティを含むそれぞれの特性 (Others) ;

それは、アカウントが (その要件と共に) 担保付き (bonded) のバリデータになる、ノミネートする、そして既存の担保付きの検証者がこのステータスからエグジットする意思を登録する。またバリデーションと正規化メカニズムのためのメカニズムも含みます。

6.2.1. ステークトークンの流動性ネットワークセキュリティをステークトークンの全体的な「時価総額」に直結させるため、一般に、できるだけ多くのトータルステキングトークンをネットワークメンテナンス操作内でステークすることが望ましい。これは、通貨のインフレを通し、収益をバリデータとして参加する人々に配ることによって、容易にインセンティブ設計することができる。しかし、そうすることは一つ問題を提起する：トークンがステークコントラクトでロックされるならば、価格向上を実現するためにどのように十分な流動性を維持できるのか？

これに対する 1 つの答えは、直接的なデリバティブコントラクトを許可し、ステイクされたトークン上で代替可能 (Fungible) なトークンを保護することだ。これは信頼できる方法で手配するのが困難だ。さらに、これらのデリバティブトークンは、異なるユーロ圏の国債が代替可能ではないのと同じ理由で同等に扱うことはできない：原資産が破綻し、価値がなくなる可能性がある。ユーロ圏の政府では、これがデフォルトとなるかもしれない。しかしバリデータがステイクしたトークンの場合、バリデータの悪意を持った行動には処罰が伴う。

私たちは信条を守りながら、最も単純な解決策を選んだ：全てのトークンはステイクされない。つまりは、ある一定の割合 (おそらく 20 % 程度) のトークンが強制的に流動性を維持することを意味する。これはセキュリティの観点からは不完全だが、ネットワークのセキュリティに根本的な違いをもたらすことはまずありえない。ステイクの没収による賠償金の 80 % と、100 % のステークの「完璧なケース」との大差はそれほどない。

ステイクされるトークンと流動的なトークンの比率はリバースオークションの仕組みによって案外簡単に決められる。基本的に、バリデーターになりた

いトークン保有者はそれぞれ、参加するために必要となる最小支払い率を記載したオファーをステーキング契約に投稿します。各セッションの開始時に（セッションは定期的に、おそらく1時間に1回程度）、各バリデーターのステークとペイアウト率に従ってバリデータースロットが満たされる。これに対する1つの可能なアルゴリズムは、目標総賭け金をスロット数で割った数以下で、その額の半分の下限を下回らない賭け金を表す最低オファーを有するものを選ぶことであろう。スロットを埋めることができない場合は、満たすために下限が何らかの要因で繰り返し引き下げられる。

ステイクしているトークンをアクティブなバリデータにトラストレスにノミネートすることは可能だ。それはバリデータに責任を任せる事になる。ノミネーションは承認投票システムによって行われる。それぞれのノミネーター志望者は、自らの責任の下で彼らがステイクを賭けるのに十分な信頼を置く、一人以上のバリデータをステークコントラクト上に登録することができる。

各セッションで、ノミネーターの掛け金は1人、またはそれ以上のバリデーターに分散される。分散アルゴリズムは、総賭け金がバリデーターセットに最適化する。(The dispersal algorithm optimises for a set of validators of equivalent total bonds.) ノミネーターの賭け金は、バリデーターの責任の下に置かれ、バリデーターの行動に応じて利益を得るか、または処罰として減額を受ける。

6.2.3. 債権の没収/バーン特定のバリデーターの振る舞いは賭け金に懲罰的な減少をもたらす。賭け金が許容最小額を下回ると、セッションは途中で終了し、別のセッションが開始される。処罰対象のバリデーターの不正行為のリストには以下が含まれる：

- パラチェーンブロックの有効性についてコンセンサスを提供できないパラチェーングループの一員である
- 無効なパラチェーンブロックの有効性について積極的に署名する
- 利用可能として以前に投票されたアウトバウンドペイロードを供給することができない
- 合意プロセス中に非活動的である
- 競合するフォークのリレーチェーンブロックを検証する

不正な悪意のある行動のケースによっては、ネットワークの完全性が損なわれ（無効なパラチェーンブロックに署名したり、フォークの複数の面を検証したりするなど）、その結果、債権の完全な没収により追い出されることがある。その他の、それほど深刻ではない不正行為（例えば、合意プロセスにおける非活動）または誰の責任か不明瞭である（無効なグループの一部であるなど）場合、代わりに、債権のごく一部の罰金を科されることがある。後者の場合、これはサブグループ churn によって、悪意のあるノードが健全なノー

ドよりより大きな損失を被るようにする。

場合によっては（マルチフォーク検証や無効なサブブロック署名など）、各パラチェーンブロックを常に検証するのは面倒な作業になるため、バリデーターがお互いの不正行為を検出することは困難である。ここでは、そのような不正行為を検証し報告するために、検証プロセス外部にある組織の支援を必要とする。その役割を担った存在はそのような活動を報告することで報酬を得る。彼らの「Fisherman（釣り人）」という名前は、そのような稀な報酬に由来している。

これらのケースは通常非常に深刻であるため、いかなる報酬も没収された債券から容易に支払うことができると我々は考えている。一般的に、大規模な再割り当てを試みるのではなく、バーン（つまり、何もしないこと）により再割り当てのバランスを取ることが好ましい。これはトークンの全体的な価値を増加させ、発見に関与する特定の関係者よりもむしろネットワークに対して補償する効果がある。これは主に安全メカニズムとしてのものである。それは大量の報酬を伴ってしまう場合、単一のターゲットに対して報告する極端なインセンティブを与えてしまう事になるかもしれないからだ。

一般的に、報酬はネットワークにとって検証を行うのに十分なほどの大きさである必要がある一方、特定のバリデーターに不正行為を強制させるような、組織化されたハッキング攻撃のインセンティブになるほど大きくないことが重要である。

このようにして、報酬は不正行為を行ったバリデーターの債権量を超えないようにする必要がある。これは、検証者が故意的に不正行為を行い、自分自身を通報する事で利益を得ないようにするためである。これへの対処法として、バリデーターになるのに最低限の賭け金を必要とすることや、ノミネーターに賭け金が少ないバリデーターは不正行為を行うインセンティブが大きい事実を啓蒙するなどがある。

3.2 6.3. パラチェーンレジストリ

各パラチェーンはこのレジストリで定義されている。それは比較的単純なデータベースのような構造であり、そして各チェーンに関する静的な情報と動的な情報の両方を保持する。

静的情報には、異なるクラスのパラチェーンを区別する手段である検証プロトコルの識別情報とともに、チェーンインデックス（単純な整数）が含まれる。これによって有効な候補を提示するために委任されたバリデーターによって正しい検証アルゴリズムが実行される。最初の POC では、新しい検証アルゴリズムをクライアント自体に配置することに重点が置かれ、追加のクラ

スのチェーンが追加されるたびにプロトコルのハードフォークが事実上必要になる。しかし最終的には、クライアントがハードフォークなしで新しいパラチェーンを効果的に処理できるように、厳密かつ効率的な方法で検証アルゴリズムを指定することが可能である。これに対する 1 つの可能な方法は、WebAssembly のような確立された、ネイティブにコンパイルされた、プラットフォームに依存しない言語でパラチェーン検証アルゴリズムを指定することである。これが本当に実現可能であるかどうかを判断するには追加の研究が必要だが、もしそうであれば、ハードフォークをしないことにより大きな利点をもたらす可能性がある。

動的情報には、パラチェーンの入力キュー (6.6. で説明) など、グローバルな合意が必要なトランザクションルーティングシステムの側面が含まれている。

レジストリは、全国民投票 (referendum) によってのみ追加されたパラチェーンを持つことができる。これは内部で管理できるが、より一般的なガバナンス要素のもとでの再利用を促進するために、外部の国民投票コントラクトに入れられる可能性が高くなる。追加チェーンの登録およびその他のあまり正式でないシステムアップグレードのための投票要件 (たとえば、必要なクォラム、大多数の要件) に対するパラメータは、「マスター規約」に記載されるが、少なくとも最初はかなり慣習的な方法に従う。正確な定式化は本研究の範囲外であるが、例えば、システムのステイクの 3 分の 1 以上が積極的に投票するという、3 分の 2 のスーパーマジョリティが賢明な出発点となるだろう。

追加の操作には、パラチェーンの一時停止と削除が含まれる。中断は決して起こらないであろうと願っているが、それはパラチェーンのバリデーションシステムに少なくともいくつかの扱いにくい問題があることを保護するセーフガードの役目を担っている。それが必要とされる可能性がある最も顕著な例は、妥当性またはブロックについて合意することができないようにバリデータを導く実装間の重大なコンセンサスの違いである。バリデータは、債権の没収前にそのような問題を発見できるようにするために、複数のクライアント実装を使用することをお勧めする。

一時停止は緊急措置であるため、国民投票ではなく動的バリデータ投票によって実行されるだろう。再検証はバリデータからも国民投票からも可能である。

パラチェーンの削除は、国民投票の後に初めて行われ、スタンドアロンチェーンへの秩序ある移行を可能にするため、または他の何らかの合意システムの一部となるためには、かなりの猶予期間が必要である。猶予期間は数ヶ月程度である可能性があり、異なるパラチェーンがそれぞれの必要性に応じて異なる猶予期間を享受できるようにするために、パラチェーンレジストリにチェーンごとで設定される可能性がある。

3.3 6.4. リレーブロックのシーリング

シーリングとは、本質的には正規化のプロセスを指す。つまり、オリジナルを意味のあるものにマッピングする基本的なデータ変換のことである。POW チェーンの下では、シーリングは事実上マイニングの同義語である。私たちの場合、それは特定のリレーチェーンブロックとそれが表すパラチェーンブロックの有効性、可用性、そして正規性に関するバリデータからの署名されたステートメントの収集を意味する。

基礎となる BFT コンセンサスアルゴリズムのメカニズムの説明は今回の範疇外となる。代わりに、合意形成ステートマシンを想定したプリミティブを使って説明する。最終的には、コアにあるいくつかの有望な BFT 合意アルゴリズム (Tangaora [9] (Raft [16] の BFT 版)、Tendermint [11]、HoneyBadgerBFT [14]) にインスパイアされることを期待している。このアルゴリズムは、複数のパラチェーンに並行して合意に達する必要があるため、通常のブロックチェーン合意メカニズムとは異なる。一旦合意に達すると、私たちはその合意を反論できない証拠として記録することができ、それは参加者の誰もが提供することができる。我々はまた、処罰に対処する際、プロトコル内の不正行為は一般に不正行為をする参加者を含む小グループにする事により、付随的な被害を最小限に抑えることができると想定している。Tendermint BFT やオリジナルの Slasher など、既存の PoS ベースの BFT コンセンサススキームは、これらの主張を満たしている。

署名付きステートメントの形式をとる証明は、リレーチェーンブロックのヘッダー、およびその他の特定のフィールド (リレーチェーンのステイトツリーのルートおよびトランザクションツリーのルート) と共に配置される。

シーリングプロセスは、リレーチェーンのブロックと、リレーのコンテンツの一部を構成するパラチェーンのブロックの両方に対応する単一の合意生成メカニズムの下で行われる。パラチェーンは、サブグループによって別々に「コミット」された後に照合されるわけではない。これにより、リレーチェーンの処理がより複雑になるが、システム全体の合意を 1 段階で完了し、待ち時間を最小限に抑え、以下のルーティング処理に役立つ非常に複雑なデータ可用性の要件を満たすことができる。

各参加者のコンセンサスマシンの状態は、単純な (2 次元の) 表としてモデル化できる。各参加者 (バリデータ) は、各パラチェーンブロック候補ならびにリレーチェーンブロック候補に関して、他の参加者からの署名付きステートメント (Vote) の形式で一組の情報を有する。情報セットは 2 つ：

- 利用可能性 (Availability)：このバリデータはこのブロックからのトランザクションの一連の情報を出力していか；故に次のブロックのパラ

チェーン候補を適切に検証できる。バリデータは1（知られている）か0（まだ知られていない）のどちらかを投票することができる。1を投票した場合、このプロセスの残りの部分についても同様に投票することに一貫する。これに従わない後からの投票は罰の対象となる。

- 妥当性 (Validity)：このパラチェーンブロックは有効であり、全ての外部参照データ（例えばトランザクション）は利用可能か。これは、投票しているパラチェーンに割り当てられているバリデータにのみ関係する。彼らは1（有効）、-1（無効）または0（まだ知られていない）のどれかに投票することができる。彼らがゼロでない (non-zero) 投票をしたら、このプロセスの残りの部分についても同様に投票することに一貫する。これに従わない後からの投票は罰の対象となる。

すべての検証者が投票を提出する必要がある。票は上記の規則によって修飾され、再提出されることがある。合意の進行は、並行して行われる各パラチェーンに対する複数の標準的な BFT 合意アルゴリズムとしてモデル化することができる。これらは少数の悪意のあるアクターが1つのパラチェーングループに集中することによって潜在的に妨害される。そのため、バックストップを確立するための全体的なコンセンサスが存在し、1つ以上の向こうパラチェーンブロックにデッドロックされる最悪のシナリオを防ぐ。

個々のブロックの有効性のための基本的な規則（はバリデータ全体が、正規のリレーから参照されるユニークなパラチェーン候補になることについて合意に達することを可能にする）：

- 少なくとも3分の2のバリデータがポジティブに投票し、誰もネガティブに投票しないこと。
- 3分の1を超えるバリデータが、外に出て行く情報の可用性にポジティブに投票している。

正当性について少なくとも1つの正と負の投票がある場合、例外条件が作成され、悪意のある当事者がいるかどうか、または偶然の分岐があるかどうかを判断するためにバリデータのセット全体が投票する必要がある。有効と無効の他に、その両方に対する投票と同等である3番目の種類の投票が許可されている。つまり、ノードには意見の対立がある。これは、ノードの所有者が同意しない複数の実装を実行していることが原因である可能性があり、プロトコルにあいまいさがある可能性があることを示している。

すべての投票が完全なバリデータセットの確認を経た後、負けた側の意見が勝った側の意見の投票のある程度の割合（パラメータ化されるために；最大で半分、おそらくかなりより少ない）を占める場合、それは偶然起こったパラチェーンフォークと考えられ、そのパラチェーンは自動的にコンセンサス

プロセスから中断される。さもなければ、それは悪意のある行為であると考えられ、反対意見に投票していた少数派を罰することになる。

結論は、正規性を示す一連の署名である。(The conclusion is a set of signatures demonstrating canonicity.) その後、リレーチェーンブロックはシーリングされ、次のブロックをシーリングするプロセスが開始される。

6.5. シーリングリレーブロックの改善このシーリング方法はシステムの運用に強力な保証を提供するが、スケールに問題があると考えられている。なぜなら、パラチェーンの重要な情報の可用性 (Availability) は、バリデータ全体の 3 分の 1 以上によって保証されている必要があるためである。そしてこれは、チェーンが追加されるにつれて、すべてのバリデータの責任範囲が増大することを意味する。

オープンコンセンサスネットワーク内のデータの可用性は本質的に未解決の問題であるが、検証ノードにかかるオーバーヘッドを軽減する方法がある。1 つ目の解決策は、バリデータはデータの可用性に対する責任を負うが、実際にデータを保存・通信・複製する必要はないことである。このデータを編集している（あるいはまったく同じ）照合者に関連している可能性がある。また、（このデータをコンパイルする照合者に関係、または同等であるかもしれない）2 次データ格納庫 (silos) が、支払いに対する利子/収入の一部を提供しているバリデータ、と可用性を保証するというタスクを管理できる。

しかしながら、これにより一時的なスケーラビリティを得られるが、根本的な問題を解決することにはならない。パラチェーンの追加は、さらなるバリデータを必要とするため、長期的なネットワークリソースの消費量（特に帯域幅の点で）はチェーンの二乗に比例して増加する。

最終的には、合計バリデータ × 合計入力情報のバンド幅の基礎制限に手の打ち止めとなるだろう。これは、信頼されていないネットワークが、他の多くのノードにデータストレージのタスクを適切に分配することができないことに起因している。

6.5.1. 待ち時間の導入この規則を緩和する 1 つの方法は、即時性の概念を緩和することだ。すぐにではなく、最終的にのみ可用性に投票する 33 % + 1 のバリデータを要求することで、指数関数的データ伝播をより有効に活用し、データ交換のピークを平準化することができる。(証明されていないが) 最も有り得そうな式は次のようになる。

(1) 待ち時間 = 参加者 × チェーン (latency = participants x chains)

現在のモデルでは、システムのサイズはチェーンの数に比例してスケールし、それにより処理が確実に分散される。各チェーンは少なくとも 1 人のバリデー

タを必要とし、可用性検証を一定比率のバリデータに固定するため、参加者はチェーンの数が増えるにつれて同様に大きくなる。そしてこれに終始する：

(2) 待ち時間 = size2 (latency = size2)

つまり、必要な帯域幅と可用性がネットワーク全体で認識されるまでの待ち時間（ファイナライズ前のブロック数とも呼ばれる）は、システムのスケールの2乗に比例して増加する。これは大きな成長要因であり、注目に値するロードロッカーになる可能性があり、「平坦ではない (non-flat)」パラダイム（リレーチェーンのツリーを介したマルチレベルルーティングを行うため、複数の「Polkadots」を階層的に構成するなど）を実現する。

6.5.2. パブリック参加もう1つの可能性のある方向性は、マイクロクレームシステムを通じたプロセスへの一般参加を許可することである。Fishermanと同様に、入手可能性を主張する検証者を監視する外部の存在がありえる。彼らの仕事は、そのような可用性を示すことができないように見える人を見つけることで、他のバリデータにミクロの苦情を申し立てることができる。システムをほとんど役に立たなくするようなシビル (sybil) 攻撃を軽減するために、電力またはステークボンドを使用することができる。

6.5.3. 可用性の保証人最終的な方法は、「可用性保証者」として2組目のステイク済みバリデータをノミネートすることである。これらは通常バリデータと同じように結合され、同じセットから取られることさえ可能だ（少なくともセッションごとに、長期間にわたって選択される）。通常バリデータとは異なり、パラチェーンを切り替えるのではなく、重要なインターチェーンデータの可用性を証明するために単一のグループを形成する。

これには、参加者とチェーン間の同等性が緩和されるという利点がある。本質的に、チェーンは（元のチェーンバリデータセットと一緒に）成長することができるが、参加者、特にデータ可用性テストに参加している参加者は、最低限の準線形 (sub-linear) に留まる可能性がある。

6.5.4. 照合者 (Collator) の設定このシステムの重要な側面の1つは、どのパラチェーン内にもブロックを作成するための健全なコレクターの選択が行われていることを確認することである。単一の照合者がパラチェーンを支配していた場合、外部データの可用性が不足する可能性はそれほど明白ではないため、いくつかの攻撃がより実行可能となる。

1つの選択肢は、擬似ランダムメカニズムでパラチェーンブロックを人為的に重み付けすることにより、照合者の多様化を行うことだ。第一に、合意メカニズムの一部として、バリデータが「より重い」と判断したパラチェーンブロック候補を支持することを要求する。同様に、バリデータが最も重いブロックを提案することを動機付ける必要がある- これは彼らの報酬の一部

を候補の重さに比例させることを通してなされるかもしれない。

照合者が彼らの候補が勝利候補として選択される合理的公平な機会が与えられることを確実にするために、パラチェインブロック候補の具体的な重みを各照合者のランダム関数で決定する。たとえば、照合者のアドレスと、作成されているブロックのポイントの近くで決定される暗号的に安全な疑似乱数の間の XOR 距離の測定（概念的な「勝利チケット」）。これにより、各照合者（より具体的には各照合者のアドレス）に、候補者ブロックが他のすべての候補者よりも「勝つ」というランダムなチャンスが与えられる。

1 人の照合者が勝利チケットに近いアドレスを「マイニング」してそのブロックをお気に入りにするというシビル攻撃を防ぐために、照合者のアドレスに慣性（inertia）を導入する。これは、アドレスに基準金額の資金があることを要求するのと同じくらい簡単かもしれない。よりエレガントなアプローチは、問題のアドレスに溜まっている資金の量で、勝利チケットの近くに重みを付けることだ。モデリングはまだ行われていないが、このメカニズムによって、ごくわずかなステイク者でも照合者として貢献できる可能性がある。

6.5.5. 太りすぎた（Overweight）ブロックバリデータセットが危険に晒されている場合、有効ではあるが実行と検証に時間がかかるブロックを作成して提案する可能性がある。バリデータグループは、ショートカットを可能にする特定の情報が既に知られていない限り、実行するのに非常に長い時間がかかるブロックを合理的に形成できるので問題となる。1 人の照合者がその情報を知っていれば、他の古いブロック処理をしている人に対し、自分の候補者を受け入れさせることに明らかなアドバンテージとなる。これらのブロックを太りすぎ（Overweight）と呼ぶ。

追加の注意点はあるが、これらのブロックを送信して検証するバリデータに対する保護は、無効なブロックとほぼ同じように考えられる。ブロックを実行するのにかかる時間は主観的であり、投票の最終結果は不正行為については、基本的に 3 つに分類される。1 つの可能性は、ブロックが明らかに太りすぎではないということだ - この場合、3 分の 2 以上がブロックをある限度内で実行できると宣言している（例えば、ブロック間の合計許容時間の 50 %）。2 つ目は、ブロックが確実に太り過ぎであるということだ。これは、3 分の 2 以上が、制限内でブロックを実行できないと宣言した場合。最後の可能性はバリデータ間の意見が半分に割れることだ。この場合、我々は何らかに比例した罰をすることを選ぶかもしれない。

バリデータがいつ太りすぎのブロックを提案している可能性があるかをバリデータが確実に予測できるようにするには、ブロックごとに自分のパフォーマンスに関する情報を公開するように要求することを勧める。十分な期間にわたって、彼らが判断しようとしている仲間と比較して彼らの処理速度をプ

ロファイルすることを可能にするはずだ。

6.5.6. コレータ保険バリデータに関して1つの問題が残っている：PoWネットワークとは異なり、有効性について照合者のブロックをチェックするために、彼らは実際にその中のトランザクションを実行しなければならない。悪意のある照合者は、無効な、または太りすぎのブロックをバリデータに供給することができ、グリーフ（リソースの無駄遣い）を引き起こし、潜在的にかなりの機会損失を強制する。

これを軽減するために、バリデータ側で単純な戦略を提案する。第一に、バリデータに送られるパラチェーンブロック候補は資金があるリレーチェーン口座から署名されなければならない。そうでない場合、バリデータはすぐにそれを削除する必要がある。第二に、そのような候補者は、ある上限までの口座内の資金の額、過去に丁寧に提案した過去のブロック数までの組み合わせ（例えば、乗算）によって優先的に順序付けされるべきである。）、および前述のように当選チケットへの近接係数。上限は、無効なブロックを送信した場合にバリデータに支払われる懲罰的損害賠償と同じでなければならない。

無効または過重のブロック候補を検証者に送信することを照合者にさせないために、検証者は不正な照合者の口座にある資金の一部または全部を不正な検証者に移すという影響で、不正行為を主張する違反ブロックを含むトランザクションを次のブロックに入れることができる。このタイプの取引は、罰金の前に照合者が資金を取り出すことをできないようにするために、他の取引を前倒して実行する。損害賠償として譲渡される資金の額は、まだモデル化されていない動的パラメータだが、生じたグリーフのレベルを反映するためのバリデータブロック報酬の割合となる可能性がある。悪意のある検証者が照合者の資金を勝手に没収するのを防ぐために、照合者は小額の入金の見返りにランダムに選択された検証者の陪審員による検証者の決定に上诉することができる。彼らがバリデータの支持を得た場合、デポジットは彼らによって消費される。そうでなければ、保証金は返却され、バリデーターは罰金を科される（バリデーターははるかにアーチ型の位置にあるので、罰金はかなり多額になるだろう）。

6.6. チェーン間トランザクションルーティングチェーン間トランザクションルーティングは、リレーチェーンとそのバリデータの重要なメンテナンスタスクの1つである。これは、転記されたトランザクションが、ある信頼要件を必要とせずに、ソースパラチェーンからの希望する出力から他のデスティネーションパラチェーンの非交渉入力になる方法を決定するロジックである。

私たちは上の言葉を慎重に選ぶ。特に、この投稿を明示的に承認したために、ソースパラチェーン内にトランザクションがあったことを要求しない。私たちのモデルに課せられる唯一の制約は、パラチェーンが全体のブロック処理

出力の一部としてパッケージされ、ブロックの実行の結果であるポストを提供しなければならないということだ。

これらのポストは複数の FIFO キューとして構成されている。リストの数はルーティングベースと呼ばれ、16 前後になる場合がある。特に、この数は、マルチフェーズルーティングに頼らなくてもサポートできるパラチェーンの数を表す。当初、Polkadot はこの種の直接ルーティングをサポートするが、最初の一連のパラチェーンをはるかに超えてスケールアウトする手段として、1 つの可能な多相ルーティングプロセス（「ハイパールーティング」）の概要を説明する。

すべての参加者が次の 2 つのブロック n 、 $n + 1$ のサブグループを知っていると仮定する。要約すると、ルーティングシステムは次の段階に従う。

- CollatorS : Contact members of V alidators $[n][S]$
- CollatorS: FOR EACH subgroup s: ensure at least 1 member of V alidators $[n][s]$ in contact
- CollatorS : FOR EACH subgroup s: assume egress $[n - 1][s][S]$ is available (all incoming post data to ' S ' from last block)
- CollatorS: Compose block candidate b for S: (b.header, b.ext, b.proof, b.receipt, b.egress)
- CollatorS : Send proof information proof $[S] = (b.header, b.ext, b.proof, b.receipt)$ to V alidators $[n][S]$
- CollatorS : Ensure external transaction data b.ext is made available to other collators and validators
- CollatorS : FOR EACH subgroup s: Send egress information egress $[n][S][s] = (b.header, b.receipt, b.egress[s])$ to the receiving sub-group' s members of next block V alidators $[n + 1][s]$
- V alidatorV : Pre-connect all same-set members for next block: let $N = \text{Chain}[n + 1][V]$; connect all validators v such that $\text{Chain}[n + 1][v] = N$
- V alidatorV : Collate all data ingress for this block: FOR EACH subgroup s: Retrieve egress $[n - 1][s][\text{Chain}[n][V]]$, get from other val- idators v such that $\text{Chain}[n][v] = \text{Chain}[n][V]$.Possibly going via randomly selected other val- idators for proof of attempt.
- V alidatorV : Accept candidate proofs for this block proof $[\text{Chain}[n][V]]$. Vote block validity
- V alidatorV : Accept candidate egress data for next block: FOR EACH subgroup s, accept egress $[n][s][N]$. Vote block egress availability; republish among interested validators v such that $\text{Chain}[n + 1][v] = \text{Chain}[n + 1][V]$.
- V alidatorV : UNTIL CONSENSUS

ここで、`egress [n] [from] [to]` は、ブロック番号 `n` の parachain `from` から parachain `to` への投稿に対する現在の出力キュー情報である。CollatorS は、パラチェーン S の Collator。Validators `[n] [s]` は、ブロック番号 `n` のパラチェーン `s` のバリデータのセット。逆に、Chain `[n] [v]` はバリデータ `v` がブロック番号 `n` に割り当てられているパラチェーン。block.egress `[to]` は、宛先 parachain が `to` である、ある parachain ブロックブロックからの投稿の出力キュー。

照合者は、自分のブロックが正規になったことに基づいて料金を徴収するため、次のブロックの送信先ごとに、サブグループのメンバーに現在のブロックからの出力キューが通知される。バリデータは（パラチェーン）ブロックについてコンセンサスを形成することのみを奨励されている。原則として、バリデータは照合者と忠誠を尽くし、他の照合者のブロックが標準的になる可能性を減らすために共謀することができるが、これはパラチェーンのバリデータをランダムに選択するため整理するのが困難であり、合意プロセスを妨げるパラチェーンブロックに対して支払うべき料金の減少で擁護される可能性がある。

6.6.1. 外部データの可用性：Parachain の外部データが実際に利用可能であることを保証することは、ネットワーク全体に作業負荷を分散することを目的とした分散型システムにおける永続的な問題である。この問題の核心は可用性の非対話的な証明も非可用性のいかなる種類の証明も行うことが出来ないため、BFT システムが正確に何らかの外部のデータの可用性に依存する遷移を適切に検証するためにシステムの許容可能なビサンチンノードの最大数 + 1 がデータが利用可能であることを述べる可用性の問題である。

Polkadot のようにシステムを適切にスケールアウトするには次のような問題が生じる。一定割合のバリデータがデータの可用性を証明しなければならず、バリデータが利用可能であると表明する前に実際にデータを保管したいと考えている場合、システムのサイズ（バリデータの数）に伴って帯域幅・ストレージ要件が増大するという問題をどのように回避すれば良いのか。その答えとして考えられるのは、別のバリデータのセット（可用性保証人）を用意することである。これについては 6.5.3 で説明する。

二次的なトリックもある。グループとして、collators(照合者) は選択したパラチェーン全てのデータが利用可能であることを保証する特有のインセンティブを持っている。なぜならデータがなければ取引手数料を収集するためのブロックを作成することが出来ないからである。collators もグループを形成するが、そのメンバーは様々で（Parachain バリデータグループのランダムな性質による）、簡単に入ることはできず、簡単に証明できる。したがって最近の collators（おそらく最後の数千ブロック）は特定の Parachain ブロックの外

部データの可用性に対する challenges を、少ない bond でバリデータに発行することができる。

バリデータは明らかに違反しているバリデータのサブグループのメンバと連絡を取り、データを取得して collator に返すか、escalate the matter by testifying to the lack of availability (データを提供することを直接拒否することは、bond を没収する犯罪であり、よって不正行為を行ったバリデータは単に接続を切断する可能性が高い)、同じテストを実行するために別のバリデータに連絡する必要がある。後者の場合は collator の bond は返される。

このような可用性のない証言を作ることができるバリデータの quorum(分散システムにおいて、分散トランザクションが処理を実行するために必要な最低限の票の数) が限界に達すると、それらは解放され、不正なサブグループは罰せられ、ブロックは元に戻される。

6.6.2 ポストルーティング。各 parachain ヘッダには” egress-trie-root” が含まれている。これは” routing-base-bins” を含む、トライのルートであり、各 bin は出力ポストの連結されたリストである。特定のパラチェーンのブロックが特定の宛先のパラチェーンの特定の出力キューを有していたことを証明するために、マークル証明をパラチェーンバリデータ全体に提供することができる。

パラチェーンブロックの処理の開始時にそのブロックにバインドされている他のパラチェーンの出力キューはブロックの入力キューにマージされる。どのパラチェーンブロックペア間にも有利性を提供しない決定論的演算を達成するために、CSPR9 サブブロック順序付けを強く仮定した。Collator は新しいキューを計算し、parachain のロジックに出力キューを排出する。

入力キューの内容は、parachain ブロックに明示的に書き込まれる。これには主に2つの目的がある。第一に parachain は他の parachain から分離して信頼できるように同期化できることを意味する。第二にもし入り口キュー全体が単一ブロックで処理できないなら、データロジスティックを単純化する。バリデータと collators は特にキューのデータソースを保持することなくブロックを処理することができる。

もし parachain の入力キューがブロック処理の終了時に閾値を超えている場合、リレーチェーンで飽和とマークされ、クリアされるまでそれ以上のメッセージはこない。マークル証明は parachain ブロック証明における collator の動作の忠実性を実証するために使用される。

6.6.3 批評この基本的なメカニズムに関連する小さな欠陥の一つは post-bomb attack である。これは全てのパラチェーンが特定のパラチェーンに可能な最大量の post を送ることである。これによりターゲットの入力キューが一度に

結び付けられるが、標準的な Dos アタック以上の被害は発生しない。

N 個の parachain、 $N \times M$ 個の合計バリデータ、および parachain ごとの L 個の collators について、よく同期された悪意のない一連の collators およびバリデータを用いて正常に動作することにより、ブロックごとの合計データ経路を以下のように分類することができる。

バリデータ: $M-1+L+L$: $M-1$ は Parachain セット内の他のバリデータ用であり、候補 parachain ブロックを提供する各 collator 用の L と、前のブロックの出力ペイロードを必要とする次のブロックの各 collator 用の第 2 の L である。後者は、実際には最悪の場合の操作に似ている。これは Collator がそのようなデータを共有する可能性が高いからである。

Collator: $M+kN$: M は関連する各 Parachain ブロックバリデータへの接続用、 kN は次のブロックの各 parachain バリデータの一部のサブセットへの出力ペイロードのシード用である。(といくつかの好ましい collators)

よってノードごとのデータ: パスの方法はシステム全体の複雑さに比例して増加する。これは理に通ってるがシステムが数百、数千の Parachain にスケールすると複雑性の増加率を低くする代わりに、ある程度の通信遅延が吸収される可能性がある。この場合、記憶バッファ及び待ち時間を導入するコストで瞬時経路の数を減少させるために、“multi-phase routing algorithm” を使用される。

6.6.4 “Hyper-cube Routing” “Hyper-cube Routing” は、上記の基本的なルーティングメカニズムの拡張として構築できるメカニズムである。本質的に parachain とサブグループの数でノードのコネクティビティを成長させるのではなく、parachain の対数でのみ成長させる。posts は最後のデリバリの途中でいくつかの parachain のキューの間を移動することができる。

ルーティング自体は決定論的で単純である。まず入力/出力の bins の数を制限するところから始める。parachain の総数ではなくルーティングベース (b) である。これは parachain の数が変化すると固定され、代わりに経路指数 (e) が上昇する。このモデルの下で、メッセージボリュームは O と共に成長し、経路は一定のままであり、O (e) と共に待ち時間（またはデリバリに必要なブロック数）がある。

このルーティングモデルは e 次元の超立方体であり、立方体の各辺は b 個の可能な位置を持つ。ブロックごとに一つの軸に従ってルーティングする。軸をラウンドロビン方式で交代させることで、e ブロックの最悪の場合のデリバリ時間を保証する。

parachain 処理の一部として、current block number (and thus routing di-

mension) を指定すると、入力キューで見つかった外部バウンドメッセージは適切な出力キューの bin に即座にルーティングされる。このプロセスはデリバリルート上の各ホップに対して追加のデータ転送を必要とするが、これはデータペイロードデリバリを必要とするが、データペイロードデリバリのいくつかの代替手段を使用し、post-trie 内のポストの全ペイロードではなく、参照のみを含むことによって緩和されえる問題である。

4 parachains、 $b=2$ 、 $e=2$ 、を使用するシステムでこのようなハイパーキューブルーティングの例を示す。

Phase 0, on each message M: • sub0: if $Mdest \in \{2,3\}$ then sendTo(2) else keep • sub1: if $Mdest \in \{2,3\}$ then sendTo(3) else keep • sub2: if $Mdest \in \{0,1\}$ then sendTo(0) else keep • sub3: if $Mdest \in \{0,1\}$ then sendTo(1) else keep

Phase 1, on each message M: • sub0: if $Mdest \in \{1,3\}$ then sendTo(1) else keep • sub1: if $Mdest \in \{0,2\}$ then sendTo(0) else keep • sub2: if $Mdest \in \{1,3\}$ then sendTo(3) else keep • sub3: if $Mdest \in \{0,2\}$ then sendTo(2) else keep

ここの二つの次元は宛先インデックスの最初の2ビットであることがわかる、最初のブロックでは上位ビットのみが使用される。2番目のブロックは下位ビットを扱う。両方が発生するとポストがルーティングされる。

6.6.5 セレンディピティの最大化。基本提案の一の変更は各サブグループに $c-1$ バリデータを有する $c-2-c$ バリデータの固定合計をみることになる。各ブロックは parachain 間でバリデータ構造の構造化されていない再分割化があるのではなく、各 parachain のサブグループの代わりに、各バリデータは次のブロックの固有の異なる parachain のサブグループに割り当てられる。これは任意の二つのブロックの間に任意の二つの parachain のペアに対して、parachain の責任を交換した2つのバリデータが存在するという不変性に繋がる。これを使用して可用性を絶対的に保証することはできないが（1つのバリデータではたとえ善意であっても時折オフラインになってしまう。）それでも一般的なケースを最適化できる。

このアプローチには合併症がないわけではない。parachain を追加するとバリデータセットの再編成も必要になる。さらにバリデータの数 n は parachain の数の2乗に結び付けられているため、最初は非常に小さく始まり、最終的には非常に早く成長し、約50 parachain の後には支持できなくなる。いずれも主要な問題ではない。最初のケースではバリデータセットの再構成はいずれにしても定期的に行う必要がある。バリデータセットのサイズに関してはあまりにも小さい場合、integer factor(整数係数) をバリデータの全体の合

計に適用して、複数のバリデータを同じ parachain に割り当てることができる。6.6.4 で議論されたハイパーキューブルーティングのような” multi-phase routing mechanism” は多数のチェーンがあるとき、多数のバリデータの必要性を軽減するだろう。

6.7 Parachain 検証：バリデータの主な目的はよく結合されたアクターとしてパラチェーンのブロックが有効であることを証明することである。これには状態遷移、含まれる外部トランザクション、入力キュー内の待機ポストの実行、及び出力キュー内の最終状態が含まれる。プロセス自体はかなり単純である。バリデータが前のブロックをシール（封印）すると次のコンセンサスのための候補 parachain ブロック候補を提供する作業を自由に開始できる。最初にバリデータは parachain collator または co-validators のいずれかを使用して、parachain のブロックの候補を見つける。parachain ブロックの候補データはブロックのヘッダ、前のブロックのヘッダ、含まれる外部入力データ (Ethereum や Bitcoin ではこのようなデータはトランザクションと呼ばれるが、原則として任意の目的のために任意のデータ構造を含むことができる。) 出力キューデータ、状態遷移有効性を証明する内部データ (Ethereum の場合は、各トランザクションの実行に必要な様々な状態/ストレージ/トライ/ノード) を含む。実証的実験では最近の Ethereum ブロックに対するこの完全なデータセットが数百 KiB であることを示している。

同時にまだ実行されていない場合、バリデータは前のブロックの遷移に関する情報を取得使用とする。最初は前のブロックのバリデータから取得し、後でデータの可用性について署名する全てのバリデータから取得する。バリデータがそのような候補ブロックを受け取ると、バリデータはそれをローカルで検証する。検証プロセスは parachain クラスのバリデータモジュールに含まれる。このモジュールは合意に依存するソフトウェアモジュールであり、Polkadot (しかし原理的には C ABI ライブラリは、単一の「参照」の実装のみ持つことから適切な安全性の低下を伴って、単一のライブラリを実装間で共有することを可能にすることができる) の実装のために作成する必要がある。このプロセスでは、前のブロックのヘッダが使用され、ハッシュを記録するために最近合意されたリレーチェーンブロックを介してその識別情報が検証される。親のヘッダの正規性が確認されると、特定の parachain クラスの検証関数が呼び出される。これは複数のデータフィールド（以前のものとほぼ同じ）を受け入れ、ブロックの有効性を宣言する単純なブール値を返す単一の関数である。

このような検証関数のほとんどは、まず親ブロックから直接派生可能なヘッダフィールドをチェックする。(例親ハッシュ、number) その後、トランザクションやポストを処理するために必要に応じて内部データ構造を設定する。Ethereum のようなチェーンでは、トランザクションの完全な実行に必要な

ノードをトライ・データベースに格納する。他のタイプには予備メカニズムがあるかもしれない。

完了すると、入力ポストと外部トランザクション（外部データが表すものであれば）が制定され、チェーンの仕様に従ってバランスが取られる。（合理的なデフォルトは外部トランザクションが処理される前に、全ての入力ポストが処理されることを要求するかもしれないが parachain のロジックが決定することである。）この制定により、一連の出力ポストが作成され、これらが collator の候補と一致することが検証される。最後に適切に設定されたヘッダが候補のヘッダと照合される。

完全に検証された候補ブロックを利用して、バリデータはそのヘッダのハッシュに投票し、必要な全ての検証情報をそのサブグループ内の co-validators に送信することができる。

6.7.1 Parachain parachain の collator は現在のブロックチェーンネットワーク上のマイナーの仕事の大半をこなす” unbonded operators” である。これらは特定の parachain に特異的である。オペレートするためにはリレーチェーンと完全に同期してあるパラチェーンの両方を維持しなければならない。

”完全同期”の正確な意味は parachain のクラスに依存するが parachain 入力キューの現在の状態は常に含まれる。Ethereum の場合、少なくとも最後の数ブロックの Merkle ツリーデータベースを維持する必要があるが、アカウントの存在を示す Bloom フィルタ、familial 情報、ログ出力、block number の逆検索テーブルなど、その他様々な可能性がある。二つのチェーンの同期を保つことに加えて、それはまた、トランザクションキューを維持し、パブリックネットワークから適切に検証されたトランザクションを受け入れることによってトランザクションを” fish” しなければならない。キューとチェーンを使用して各ブロックで選択されたバリデータの新しい候補ブロックを作成し（リレーチェーンが同期化されているので、アイデンティティが分かっている）proof-of-validity などの様々な補助的情報と共にピアネットワーク経由で送信できる。トラブルのために含まれている取引に関する全ての手数料を徴収する。この仕組みには様々な経済学が導入されている。過度に競争の激しい市場では、collators の余剰性がある場合、特定の collators のブロックを含めるインセンティブを与えるために、取引手数料を parachain バリデータと共有することができる。同様に collators の中にはブロック検証者にとって魅力的なものにするために支払う必要のある料金を引き上げる場合もある。この場合自然市場が形成され、より高い手数料が列に並ばず、より早くチェーンに含まれるべきである。

6.8 Networking Ethereum や Bitcoin のような伝統的なブロックチェーン上のネットワークはかなり単純な要件を持っている。全てのトランザクション

とブロックは、undirected gossip でブロードキャストされる。同期化は特に Ethereum との同期化より複雑であるが、実際にはこのロジックにはいくつかの要求及び応答メッセージタイプを解決するプロトコル自体ではなく、ピア戦略に含まれていた。

Ethereum は単一のピア接続上で多数のサブプロトコルが多重化することを可能にし、同じピアオーバーレイが多数の p2p プロトコルを同時にサポートする devp2p プロトコルを使用して、現在のプロトコルオフリングに進化したが、プロトコルの Ethereum 部分は比較的単純なままであり、p2p プロトコルは QoS サポートなどの重要な機能が欠落したままの未完成のものである。残念なことにもっとユビキタスな” web3” プロトコルを作ろうという試みは大失敗に終わり、それを使っている唯一のプロジェクトは Ethereum のクラウドセールから明示的に資金提供されたものだった。Polkadot の必要条件はもっと重要だ。完全に統一されたネットワークではなく、Polkadot にはピア構成に対して異なる要件を持つ複数のタイプの参加者と、特定のデータについて conversing(会話する) 傾向のある複数のネットワーク” avenues” (大通り) がある。これは実質的により構造化されたネットワークオーバーレイ (及びそれをサポートするプロトコル) が必要になる可能性が高いことを意味する。さらに新しい種類の” chain” のような将来の追加を用意にする拡張性は、それ自体新しいオーバーレイ構造を必要とする可能性がある。ネットワークプロトコルがどのようなようになるかの詳細な議論はこのホワイトペーパーの範囲外であるが、いくつかの必要条件の分析は合理的である。ネットワーク参加者を大まかに3つの部分集合のそれぞれ二つの集合 (リーレーチェーン、パラチェーン) に分けることができる。parachain の参加者は他の parachain の参加者とは対照的に、自分自身の間の conversing(会話) にのみ関心があると述べることもできる。

- Relay-chain participants:
- Validators: P , split into subsets $P[s]$ for each parachain
- Availability Guarantors: A (this may be represented by Validators in the basic form of the protocol)
- Relay-chain clients: M (note members of each parachain set will also tend to be members of M)
- Parachain participants:
- Parachain Collators: $C[0], C[1], \dots$
- Parachain Fishermen: $F[0], F[1], \dots$
- Parachain clients: $S[0], S[1], \dots$
- Parachain light-clients: $L[0], L[1], \dots$

一般にこれらのセットのメンバー間で発生する傾向のある特定のクラスの通信には次のようなものがある。

- $P|A \leftrightarrow P|A$: The full set of validators/guarantors must be well-connected to achieve consensus.
- $P[s] \leftrightarrow C[s] \mid P[s]$: Each validator as a member of a given parachain group will tend to gossip with other such members as well as the collators of that parachain to discover and share block candidates.
- $A \leftrightarrow P[s] \mid C \mid A$: Each availability guarantor will need to collect consensus-sensitive cross-chain data from the validators assigned to it; collators may also optimise the chance of consensus on their block by advertising it to availability guarantors. Once they have it, the data will be disbursed to other such guarantor to facilitate consensus.
- $P[s] \leftrightarrow A \mid P[s']$: Parachain validators will need to collect additional input data from the previous set of validators or the availability guarantors.
- $F[s] \leftrightarrow P$: When reporting, fishermen may place a claim with any participant.
- $M \leftrightarrow M \mid P \mid A$: General relay-chain clients disburse data from validators and guarantors.
- $S[s] \leftrightarrow S[s] \mid P[s] \mid A$: Parachain clients disburse data from the validator/guarantors.
- $L[s] \leftrightarrow L[s] \mid S[s]$: Parachain light clients disburse data from the full clients.

効率的な転送メカニズムを保証するには、Ethereum の devp2p のように、各ノードがピアの fitness を区別しないフラットなオーバーレイネットワークが適しているとは考えられない。適切なピアが適切なタイミングで”serendipitously”(偶然に) “接続されることを保証するために、積極的に先読み計画するだけでなく、適度に拡張可能なピア選択及び検出メカニズムをプロトコル内に含める必要がある。

適切にスケールアウトされたマルチチェーンの場合、collators はそれに応じて選択されたバリデータに継続的に再接続する必要があるか、バリデータのサブセットと継続的な合意を必要とし、バリデータが自分自身にとって役に立たない大部分の時間の間、それらが切断されないことを保証する必要がある。また collators は可用性保証セットへの一つ以上の安定した接続を維持し、consensus-sensitive データの迅速な伝播を保証しようとする。

可用性保証の主な目的は、相互の安定した接続、バリデータ（彼らが証明する合意と合意に不可欠な parachain データのために）collators(parachain データ用)、fisherman,full client(情報拡散のために) を維持することである。バリデータは他のバリデータ、特に同じサブグループにあるバリデータや、parachain ブロック候補を提供できる任意の collators を探す必要がある。

fisherman は一般的なリレーチェーンや parachain client と同様に一般的にバリデータや保証人に接続を開いたままにしておくことを目的とするが、それ以外の点では自分と似た多くのノードがある。parachain light client は他の parachain light client だけでなく、parachain full client に接続されることを同様に目的とする。

6.8.1 Peer Churn の問題 * churn<-掻きまわす基本プロトコル提案では、これらのサブセットの各々は、parachain 遷移を検証するために割り当てられたバリデータがランダムに選択されるので、各ブロックで絶えずランダムに変化する。これは異なる（ピアでない）ノード間でデータを渡す必要がある場合に問題になる必要がある。hop-distance(したがって最悪の場合の待ち時間)がネットワークサイズ（ここでは Kademlia-like プロトコルが役に立つかもしれない。）の対数でのみ増加することを保証するために、公平に分散され well-connected ピアネットワークに頼るか、ノードの現在の通信ニーズを反映するピアセットを維持するために必要な接続ネゴシエーションを行うためにより長い block times を導入しなければならない。

いずれも優れた解決策ではない。ネットワークに長時間の block time がかかると特定のアプリケーションやチェーンでは使用できなくなる可能性がある。完全に公平で接続されたネットワークであっても、関心のないノードが不要なデータを転送しなければならないような拡張が進むと、帯域幅はかなり無駄になる。

両方向が解決の一部を形成する可能性があるが、待ち時間を最小にするのに役立つ合理的な最適化は、一連のブロックの間だけメンバーシップを再割り当てする（例えば、15 のグループでは 4 秒の block time では、接続の変更は毎分一回のみであることを意味する。）かメンバーシップを増分的に回転させる（例えば、各 parachain に 15 のバリデータが割り当てられているとすると、平均して完全にユニークなセットの間は 1 分になる）ことによって、これらの parachain バリデータセットのボラティリティを制限することである。”peer churn” の量を制限し parachain セットの部分的な予測可能性を通して有利なピア接続が十分に確立されていることを保証することによって各々ノードが永続的に予期せぬピア選択を維持するのを助けることができる。

6.8.2 有効なネットワークプロトコルへのパスおそらく最も効率的で合理的な開発努力は、独自のプロトコルを展開するのではなく、既存のプロトコルを利用することに集中することである。Ethereum 独自の devp2p、IPFS の libp2p 及び GNU の GUNet を含む、いくつかの p2p ベースプロトコルがある。これらのプロトコルと特定の構造的な保証、動的な peer steering、拡張可能なサブプロトコルをサポートするモジュラーピアネットワークを構築するためのそれらの関連性の完全なレビューはこのホワイトペーパーの範囲を

はるかに超えているが、Polkadot の実装の重要なステップである。

4 7. プロトコルの実用性

4.1 7.1. チェーン間のトランザクション支払い

Ethereum のガスのような、全体的な計算リソースの必要性を無くすフレームワークによって、大きな自由と単純さが得られる一方で、これは重要な問題を提起する：いかにパラチェーン A はパラチェーン B による計算の強制を防ぐのか。1つのチェーンがトランザクション”データ”を別のチェーンにスパムすることはトランザクションポスト入力キューバッファ (transaction-post ingress queue buffers) によって防ぐことができるが、トランザクション”処理”のスパムを防ぐためのメカニズムはプロトコルによって提供されていない。

これはより高レベルの問題である。チェーンは送られてくるトランザクション POST データに任意のセマンティクスデータを付けることができるため、料金は計算を開始する前に確実に支払われる必要がある。Ethereum Serenity によって支持されたモデルと同様に、一定量の処理リソースの提供と引き換えに、バリデータに保証された支払いを可能にする”break-in”コントラクトをパラチェーン内で想像することができる。これらのリソースは、ガスのようなもので測定されるかもしれないが、主観的な実行時間や Bitcoin のようなフラットフィーモデルのような、まったく新しいモデルである可能性もある。

“break-in”コントラクトで認識されている価値のメカニズムが何か、オフチェーンの呼び出し元からは想定できないため、これ単体ではそれほど役に立たない。しかし、ソースチェーンに二次的な”break-out”コントラクトがあると想像できる。この2つのコントラクトは互いに橋渡しをし、お互いを認識し、価値の同等性を提供する（それぞれが持つステイクトークンが balance-of-payment を行うのに使われるかもしれない）。他のそのようなチェーンに Call することは、このブリッジを介してプロキシを行うことを意味する。これは、目的地のパラチェーンに必要な計算リソースを支払うために、チェーン間の価値の移転を交渉するための手段を提供する。

4.2 7.2. 追加のチェーン

パラチェーンの追加は比較的安価な操作だが無料ではない。パラチェーンが多いほど、パラチェーンごとのバリデータが少なくなり、最終的には平均賭け金量が少なくなったバリデータが多くなる。パラチェーンを攻撃するためのコスト減少問題は、Fisherman の役割によって軽減されるが、成長しているバリデータセットは、元となるコンセンサスメカニズムにより長い待ち時

間を発生させる。さらに、各パラチェーンは、過度に煩わしい検証アルゴリズムを使用して検証者を苦しめる可能性をもたらす。

よって、バリデータやステーキングしているコミュニティのために、新しいパラチェーンを追加することに値段が付いている。この「チェーン市場」では、次のいずれかが追加される可能性がある：

- （ステーキングトークンのロックアップやバーンの観点から）メンバーになるために拠出金の支払いがゼロになっている一部のチェーン（例：コンソーシアムチェーン、Doge チェーン、アプリ固有のチェーン）。
- 他では手に入れることが困難な特定の機能（機密性、内部スケーラビリティ、サービスの結びつきなど）を追加することにより、ネットワークに固有の価値を提供するチェーン。

基本的には、ステイク保持者のコミュニティに子供チェーンを追加してもらうため、インセンティブ（例：金銭・機能的なチェーンをリレーに追加したいという意味）を与える必要がある。追加された新しいチェーンに対しての削除通知の期間は非常に短い。これにより、中長期的な全体の価値へのリスクなしに、新しいチェーンを実験的に追加することができるようになることを想定している。

#8. 結びここまで、スケーラブルで特定の既存ブロックチェーンネットワークと下位互換性がある、異多種を含むマルチチェーンプロトコルをいかに実現するかを概説した。そのようなプロトコルの下で参加者は、例外的に自由に拡張することができる、標準的なブロックチェーン設計から来る既存のユーザにも特に費用が要さない、システムを創り上げるために自己関心で行動する。すでに参加者の性質、参加者の経済的インセンティブ、参加するために必要なプロセスなど、それに伴うアーキテクチャの大まかな概要を示した。基本デザインを分析し、その長所と短所について説明した。これからの我々の方向は、これらの短所・制限を減らし、完全にスケーラブルなブロックチェーンソリューションを生み出すことを目指す。

4.3 8.1. 欠けている資料と未解決の質問

ネットワークフォークの可能性は、プロトコルの異なる実装によっては常に存在する。そのような例外的な状態からの回復についての議論はまだ行われていない。ネットワークには必ずゼロではないファイナライズ期間があることを考えると、リレーチェーンのフォークから回復することは大きな問題ではないはずだが、合意プロトコルへの慎重な統合が必要になる。

債券の没収と通報による報酬の提供は、まだ深く検討されていない。現時点では、報酬は winner-takes-all を元に提供されると想定している。これは、

Fisherman にとって最良のインセンティブモデルではないかもしれない。短期間のコミット啓示プロセスでは、多くの Fisherman が報酬のより公平な配分を主張することができるが、そのプロセスは不正行為の発見にさらなる遅れをもたらす可能性がある。

4.4 8.2. 謝辞

これを漠然とした形にするのを手伝ってくれたすべての校正読者に感謝します。特に、Peter Czaban、Björn Wagner、Ken Kappler、Robert Habermeier、Vitalik Buterin、Reto Trinkler、Jack Petersson。特に Marek Kotewicz 氏と Aeron Buchanan 氏は特別な感謝が必要です。他にもアイデアを提供してくれたすべての人々と、その始まりに感謝します。そして、その過程での彼らの助けを他の皆に感謝します。すべての誤りは私の責任とします。

合意アルゴリズムの初期調査を含むこの研究の一部は、Innovate UK プログラムの下で、英国政府によって部分的に資金提供されました。

4.5 APPENDIX A. 機能コンポーネント

ハイレベルから見ると、Parity Polkadot Platform スタックには多数の機能コンポーネントがあり、その完成にはかなりの量の研究開発が必要になる。コンポーネントには既存の他コンポーネントに依存しているものも、独立しているものもある。プラットフォームが正しく機能するために非常に重要なものもあれば、あったほうがいい (nice-to-haves) ものものもある。未確定の複雑さでまだ実行可能と見なされていないものや、比較的簡単なものもある。ここでは、それらのコンポーネントをできる限り多く、それらが開発ロードマップのどこに収まるかをリストする。

- ネットワーキングサブシステム (Networking subsystem)：ピアネットワークが形成され維持される手段。初期のシステムでは、既存の p2p ネットワークライブラリ (devp2p) に単純な変更をするだけで十分である。ただし、ネットワークが成長するにつれて、ネットワーク topology がますます構造化され、最適なデータロジスティクスが可能になるようにするには、追加の研究開発が必要になる。最終的なデプロイのために、libp2p、devp2p、および GNUnet の適応は最初に考慮されるべき問題だ。要件が満たされそうにない場合は、新しいプロトコルを検討する必要がある。
- コンセンサスメカニズム (Consensus mechanism)：Proof-of-authority
コンセンサスメカニズム: 豊富なバリデータステートメントをサポートし、部分的なステートメントの主観的な受領に基づき、複数の独立した

アイテムを一連の単一プロセスの下で合意を可能にする。このメカニズムは、悪意のあるバリデータを棄却するための不正行為の証明を許可する必要がありますが、ステーキングメカニズムを含む必要はありません。相当量の研究とプロトタイピングがこのコンポーネントの開発に必要。

- Proof-of-stake チェイン：合意メカニズムを POS 領域に拡張する。このモジュールには、ステイクトークン、検証者プールへの出入り管理、検証報酬を決定する市場メカニズム、ノミネーション承認投票メカニズムのファイナライズ、および債券の没収、削除の管理が含まれる。まだ最終的な開発の前に相当量の研究とプロトタイプが必要。
- Parachain の実装 (Parachain Implementation)：最初の Parachain の実装は、Bitcoin や (より豊富なトランザクションを提供するため) Ethereum などの既存のブロックチェーンプロトコルに大きく基づく可能性がある。これには、POS チェーンとの統合が含まれ、パラチェーンが独自の内部コンセンサスメカニズムなしに合意を獲得することを可能にする。
- トランザクション処理サブシステム (Transaction processing subsystem)：パラチェーンとリレーチェーンの進化により、トランザクションの送信、受信、伝播が可能になる。これには、ネットワーク層でのトランザクションキューイングと最適化されたトランザクションルーティングの設計が含まれる。トランザクションルーティングサブシステム：これはリレーチェーンにさらなる複雑性をもたらす。まずは、パラチェーンに取引可能性 (transactability) を追加することが必要になる。それに続き、リレーチェーンにハードコードされた 2 つのパラチェーンで、それは入力/出力キューの管理を含む。最終的に、ネットワークプロトコルとともに有向トランザクション伝播 (directed transaction propagation) の手段へと発展し、独立したパラチェーン照合者が興味のないトランザクションに過度にさらされないようにする。
- リレーチェーン (Relay chain)：これはリレーチェーン (relay-chain) の最終段階で、パラチェーンの動的な追加、削除、緊急停止、不正行為の報告、そして「Fisherman」機能の実装を含みます。
- 独立した照合者 (Independent collators)：これは代替のチェーン固有の照合者機能の提供である。証明作成 (照合者用)、パラチェーンの不正行為検知 (fisherman 用)、検証機能 (検証者用) が含まれる。また、2 人が発見して通信するために必要な追加のネットワークも含まれる。
- ネットワークダイナミクスのモデル化と研究：このプロトコルの全体的なダイナミクスは徹底的に研究されるべきである。これは、オフラインのネットワークモデリングと、シミュレートされたノードを介した実証的根拠の両方によって起こり得る。後者はリレーチェーンに依存しており、ノードが照合者のために彼らのアクションに関する詳細なレポート

を中央ハブに提出することを可能にする、構造化されたロギングプロトコルの開発を含む。

- ネットワークインテリジェンス：複雑な分散型マルチパーティネットワークとして、<http://ethstats.net> に似たネットワークインテリジェンスハブがネットワーク全体のライフサインを監視し、潜在的な破壊行動にフラグを立てるために必要である。最大の効率を得るためには、構造化ロギングを使用し、このデータをリアルタイムで生成、および視覚化する必要がある。そしてそれは、リレーチェーンが合理的な完全状態であることに依存する。
- 情報公開プラットフォーム (Information publication platform)：これは、そのブロックチェーンに関連するデータを公開するためのメカニズムであり、事実上、分散型コンテンツ発見ネットワークを意味する。最初は基本的な P2P 検索で処理できるが、可用性が重要であるため、デプロイにはより構造化された堅牢なソリューションが必要になるだろう。IPFS 統合は、これらの目標を達成するための最も賢明な手段かもしれない。
- Javascript インタラクションバインディング：ネットワークとインタラクトするための主な手段は、おそらく Ethereum の例に従うだろう。そのため、高品質の Javascript バインディングが重要である。私たちのバインディングは、リレーチェーンと最初のパラチェーンとの相互作用をカバーし、それ自体はそれらに依存する。
- ガバナンス：当初、これはハードフォーク、ソフトフォーク、およびプロトコルの再パラメータ化などの例外的なイベントを管理するためのリレーチェーン上のメタプロトコルになる。それはコンフリクトを管理し、ライブロック (live-locks) を防ぐための近代的な構造である。最終的に、これは通常、ハードフォークにしかできない変更を実行することができる完全なメタプロトコル層になるかもしれない。リレーチェーンが必要。
- インタラクションプラットフォーム：ノーマルユーザーが、賭けプロセスへの参加、投票、トークン転送、ノミネーター、バリデータ、Fisherman、または照合者になるなどの一般的なタスクを容易にするための、最小限の機能を用いてシステムと対話できるプラットフォーム。機能的なリレーチェーンを持つことに依存する。
- ライトクライアント：開発されるリレーチェーンとあらゆるパラチェーンのためのライトクライアント技術。これにより、クライアントは、トラストレスに、ストレージや帯域幅をほとんど必要とせず、チェーン上のアクティビティに関する情報を入手できるようになる。リレーチェーンに依存する。
- パラチェーン UI：マルチチェーン、マルチトークンのウォレットと ID

管理システム。ライトクライアント技術とインタラクションプラットフォームが必要。これは初期のネットワーク配置には必要ない。

- チェーン上の Dapp サービス：初期のパラチェーンにデプロイする必要があるさまざまなサービス（API、名前、自然言語の仕様、コードなどのための登録ハブなど）。これはパラチェーンによるが、初期のネットワーク配置には必ずしも必要ではない。
- アプリケーション開発ツール：開発者を支援するために必要なさまざまなソフトウェアが含まれている。例としては、コンパイラ、キー管理ツール、データアーカイバ、VM シミュレータなど、他にもたくさん存在している。これらは必要に応じて開発する必要がある、そのようなツールの必要性を最小限に抑えるために、テクノロジーは部分的に選択される。またその多くは厳密には必要とされていない。
- パラチェーンとしての Ethereum (Ethereum-as-a-parachain)：Ethereum から/へのトラストフリーのブリッジにより、投稿されたトランザクションがパラチェーンから Ethereum ヘルディングされ（他の外部からのトランザクションと同じ扱い）、Ethereum 上のコントラクトがパラチェーンとその内部のアカウントにトランザクションを送れるようにする。最初に、これは実現可能性を確かめるためにモデル化され、合意プロセスに必要なバリデータの数、それが依存している要素に基づき、構造上の制限を確定する必要がある。これに続き、Proof-of-concept を構築することができ、最終的な開発はリレーチェーン自体に依存する。
- Bitcoin-RPC 互換レイヤー (Bitcoin-RPC compatibility layer)：リレーチェーン用の単純な RPC 互換レイヤーで、そのプロトコルを使用して既に構築されているインフラを Polkadot と相互運用可能にし、サポートの手間を最小限に抑える。リレーチェーンを必要とする挑戦的な目標。
- Web 2.0 バインディング：Polkadot インフラをレガシーシステムで使いやすくするために、共通の Web 2.0 テクノロジスタックにバインドします。最初のパラチェーンと公開されるチェーン上のインフラに依存する挑戦的な目標。
- zk-SNARK パラチェーンの例：zk-SNARK を利用して、トランザクションの身元を確実に秘匿にするパラチェーン。挑戦的目標はリレーチェーンに依存します。
- 暗号化されたパラチェーンの例：状態の各項目を暗号化された、および署名された状態に保つパラチェーン。これらは、その中のデータの検査と修正に対するアクセス制御の執行を確実にし、商業的に規制された業務が必要に応じて順応することを可能にする。元となる情報が晒されることなく、Polkadot バリデータがある程度の状態遷移の正しさを保証することを可能にするための Proof-of-authority メカニズムを含む。リレーチェーンに応じた挑戦的な目標。

- **トラストフリーな Bitcoin ブリッジ**：トラストフリー Bitcoin の「双方向ペグ (two-way-peg)」機能。これは、おそらく、閾値署名、あるいは SPV 証明&専門の Fisherman と一緒に $m / n(n \text{ of } m)$ マルチ署名 Bitcoin アカウントを使用するだろう。開発は初期の実現可能性分析に基づいており、好ましい結果が得れている。この機能をサポートするには、Bitcoin プロトコルに/から追加機能/ロック解除が必要。
- **抽象的/低レベルの分散型アプリケーション (Abstract/low-level decentralized application)**：トラストフリートークン交換、資産追跡インフラ、クラウドセールスインフラ。
- **コントラクト言語**：プロジェクトに絶対必要な部分ではないが、挑戦的な目標だといえる。チュートリアル、ガイドライン、および教育ツールとともに安全なコントラクト言語が用意されるだろう。それはオリジナル Solidity 言語のビジョンに従い、形式的な証明をする手段となるか、あるいは関数型言語や条件付き言語のような重大なプロセスエラーを最小にするプログラミングパラダイムに基づいているかもしれない。
- **IDE**：コントラクト言語に基づき、これはパラチェーン上でのコントラクトの編集、コラボレーション、発行およびデバッグを容易にする。さらなる挑戦的目標である。

5 APPENDIX B. FREQUENTLY ASKED QUESTIONS

- Polkadot は (ブロックチェーン名) を置き換えるように設計されているか？ : いいえ。Polkadot の目標は、新しいブロックチェーンを作成し、既存のブロックチェーンを移行することができるフレームワークを提供することである。
- Polkadot は (暗号通貨名) を置き換えるように設計されているか？ : いいえ。Polkadot トークンは通貨として使用されることを意図されたものでも、デザインされたものでもない。通貨としてデザインされた場合、その大部分はステイク制の中で非流動的となり、流動的な部分は所有権の移転をする際などに高額な手数料を発生させるだろう。それよりも、Polkadot トークンの目的は、Polkadot ネットワークのステイクを直接的な表現することにある。
- Polkadot ステイキングトークンのインフレ率はいくらか？ : Polkadot ステイキングトークンの基本拡張は無制限である。検証プロセスで長期債権として保有されているトークンの特定の割合をターゲットにするために、市場の効果に従って増減する。
- トークンの所有権を賭けることがステークホールディングを反映するのはなぜか？ : これは、ネットワークのセキュリティを支えるものであ

るという事実によって実現されるメカニズムである。そのため、それらの価値は、Polkadot が提供する全体的な経済的価値に結び付けられている。Polkadot が正常に動作していることから全体的な価値を獲得したすべてのアクターは、それが確実に継続されるように行動するインセンティブを得る。そのための最善の方法は、検証プロセスに参加することであり、これは通常、ステイクトークンの所有権を意味する。