# ブロックチェーンの課題と解決策の進展に基づく展望の考察

Overview the Blockchain Technology from Its Challenges and Solutions

秋谷 昂志†

Takayuki Akiya

フリーランス

† Email：noir.entith.marge@gmail.com

**概要　ブロックチェーンは、暗号理論を主とする技術を用いてトラストレスに P2P ネットワークでデータの遷移を記録する技術である。ブロックチェーンの直面する代表的な課題の一つにスケーラビリティがある。ビットコインのような従来の実装では十分なトランザクションを処理できず、実用上の課題があるとされている。オンチェーンのプロトコルの変更により解消を試みることもできるが、一般にネットワークのノードに負荷を転嫁する結果となるため限界がある。そこで、ペイメントチャネルやサイドチェーン等のオフチェーンスケーリング手法の研究が盛んに行われている。サイドチェーンでは高速コンセンサスアルゴリズムの採用により、トレードオフはありながらもトランザクション処理量を大きく向上させている。スケーラビリティ以外の課題も多くあり、必ずしもブロックチェーンが有効に作用しない分野もあるが、データやロジックの記述・不可逆的な記録・状態遷移の追跡といった普遍的な性質と実用性の向上が相まって、今後の応用や新分野の開拓が期待される。**

## 1.はじめに

ブロックチェーンは、新分野の開拓や既存の多様な分野への応用が可能なポテンシャルを秘めた技術である。始まりは、2008 年に Satoshi Nakamoto が発表した論文「Bitcoin: A Peer-to-Peer Electronic Cash System」に遡る。Satoshi は、信頼できる第三者機関が介在する電子決済システムの弱点を指摘し、暗号理論による証明をベースとする無与信下でピアトゥピアネットワークを介して行う二者間の電子決済システムを提案した[1]。この論文には、「ブロック（block）」や「鎖状につなぐ（chain of ～）」という現在のブロックチェーンの根幹を成す概念が散りばめられている。一方で、いまや名詞としてよく知られているブロックチェーン（blockchain）は、実は一度も使われていない。10 年の歳月を経る間に、ブロックチェーンという名詞にまで昇華されて市民権を獲得するとともに、その応用はもはや決済だけに留まらないものとなった。研究範囲は広範に及び、進展も速い。新技術の開発だけでなく、良く知られているものからニッチなものまでを含む広範な既存技術を組み合わせて次々と進歩している。

しかし、一般には暗号通貨（crypto currency）の文脈からブロックチェーンを捉えることが多いように思う。確かに暗号通貨を実現する技術であるという認識は誤りではないし、むしろ経済的価値を有するアセットを発行できることはブロックチェーンの非常に重要な側面である。だが、これはブロックチェーンのあくまで一側面であり、他の有力な側面に関しては課題や解決法も含めてあまり理解されていないように見受けられる。

筆者は、技術の進展速度に加えて暗号通貨の部分が大きく捉えられがちである点が、ブロックチェーンの理解を難しくしている理由の一つだと考えている。そこで本稿ではブロックチェーンの代表的な課題としてスケーラビリティ問題を取り上げ、解決法の進展について概観する。そのうえで、現在のブロックチェーンにどのような特徴があるのか踏まえて展望を説明することで技術としてのブロックチェーンに対する見識を深めることを目的とする。

## 2.ブロックチェーンとは

### 2.1.用語の定義

「ブロックチェーン」を、ビザンチン障害を含む不特定多数のノードを用い、時間の経過とともにその時点の合意が覆る確率が 0 へ収束するプロトコル、またはその実装と定義する。これは日本ブロックチェーン協会の定義に基づく[2]。少々難解に思われるので、ブロックチェーンの構成要素やアクターについて簡単に定義したあと、説明する。

ブロックチェーンの主な構成要素が「トランザクション」と「ブロック」である。「トランザクション」は、あるユーザから他ユーザへの価値の移転を主な成分として含む構造体である。「ブロック」は、自身と直前のブロックを説明する情報並びにいくつかのトランザクションを主な成分として含む構造体と表現できる。このようにブロックが直前のブロックを参照する構造が継続するため総体として鎖状に見えることからブロックチェーンと形容される。

ブロックチェーンの維持に関わるネットワークが「P2P ネット

ワーク（ピアトゥピアネットワーク）」である。P2P ネットワークは不特定多数の「ノード」からなる。ノードはブロックチェーンにおいてブロックの生成やメッセージの伝達等を含む複数の役割を担う。また、ブロックチェーンそのものの複製を保持する。このようなノードを特にフルノードという。文脈によってはノードをピア、バリデータ、マイナーなどと表現する方が正確であるが、簡単のためノードで統一する。ノードがブロックを生成する平均的な時間間隔を「ブロックタイム」といい、ブロックのデータサイズを「ブロックサイズ」という。生成されたブロックの承認などのように、複数のノードで構成される P2P ネットワークがある値について合意するためのルールを「コンセンサスアルゴリズム」という。

ブロックやブロック内に含まれているトランザクション対してなされた合意が覆らない確率を「ファイナリティ」と呼ぶ。通常、時間が経つにつれてファイナリティは高くなる。つまり、古いブロックほど覆りにくくなる。ファイナリティをある・なしに二元論的に分類する考え方、すなわち原理的に 100%覆らないものだけをファイナリティがあるとし、それ以外をファイナリティがないとする考え方もあるが今回は確率的に捉える。「ファイナリティ」が確率的である場合（二元論的な捉え方ではファイナリティがない場合）、任意の時点で 2 つ以上の有効なブロックが出現して承認されるとブロックチェーンに「分岐」が発生する。ブロックチェーンでは原則的に長い方が正とプロトコルで定義されており、短いブロックチェーンは破棄されるので最終的には分岐は解消され、1 つのチェーンだけが生き残る。破棄されたチェーンにおける分岐点となったブロックよりあとに生成された全てのブロックは無効となるので、当該ブロック内のトランザクションも無効になる。

ブロックチェーンは不特定多数のノードに維持を任せているため、そのふるまいを制御することは難しいが、非常に重要である。そこで、ネットワークの障害モデルとしてノードが作為か不作為かによらず偽の情報を伝達する可能性がある、すなわち「ビザンチン」なノードがいる状況を仮定して設計される。偽の情報の伝達とは、例えば故障による不応答やネットワークの非同期による古い値のブロードキャスト等も含まれる広い概念である。同時に、できる限りノードに正しいふるまいを促すべく、正しいふるまいにはインセンティブを、不正なふるまいにはペナルティを与えるなどのインセンティブスキームが組み込まれている。それらは通常、ブロックチェーンが発行するコインなどの「アセット」の形式をとる。

翻って、ブロックチェーンの定義を見てみる。まず、「ビザンチン障害を含む不特定多数のノード」について説明する。これは、ネットワークの構成が、データを管理する主体がおりその主体を介してデータのやりとりを行うクライアントサーバ方式ではなく、管理主体を持たず直接データをやりとりするピアトゥピア方式であることを意味する。そして、そのネットワークはビザンチンなノードによって構成され、ノードは必ず正しい値を伝達するとは限らない。次に、「時間の経過とともにその時点の合意が覆る確率が 0 に収束する」について説明する。これは、合意の形成に至るまでの手順であるコンセンサスアルゴリズムに従ってノードが合意形成し続ける限りにおいては、その累積によって過去の合意の変更または削除の成功確率が限りなく 0 に近づくことを指す。今回の定義では、ブロックチェーンの本来の機能と考えてよい。最後に「その実装」であるが、ノードが機能を果たすのに必要な実装、またはその実装に用いる技術群と捉えればよい。

## 2.2.基本的な構造

ブロックチェーンは暗号化や電子署名、ハッシュ化といった暗号理論のテクノロジを駆使する。暗号化は平文をある規則に従って暗号文に変換するものである。規則を知っている者であれば暗号文を平文に復号できるが、それ以外の者には復元できないため情報が秘匿される。暗号化と復号の鍵が同じものを対称暗号（共通鍵暗号）、異なるものを非対称暗号（公開鍵暗号、公開鍵と秘密鍵のペアを持つ）という。電子署名は公開鍵暗号の応用例の一つで、メッセージの作成者が確かにその人であることと改竄されていないことを保証するものである。ハッシュ化は一方向関数を用いてデータの置換・要約を不可逆的に行うものである。入力値が少しでも違うと全く異なる出力値が得られるが、出力値のデータサイズは入力値によらず一定であるという性質を持つ。これらの暗号理論によりブロックどうしを次々とつないでいくことで、時間の経過とともにその時点の合意が覆る確率を 0 へと導く。具体的にどのように技術が用いられているかのイメージを掴むことは次章以降で述べる課題やその解決策を理解するうえで有用であるので、ブロックチェーンとしては最も基本的なビットコインを取り上げて構造を簡潔に説明する。

ビットコインの利用は、基本的にビットコインウォレットと総称されるソフトウェアを用いて、秘密鍵と公開鍵のペアを生成するところから始まる。これらは主に電子署名のために用いられる。ビットコインにおける電子署名は楕円曲線暗号 secp256k1 に基づく楕円曲線電子署名アルゴリズム（ECDSA）で、巨大な乱数を

秘密鍵として secp256k1 から公開鍵を求める。また、公開鍵はユーザのアドレスとなり、このアドレスを用いて送金の受け取りができる。より詳しくは、公開鍵を SHA-256 でハッシュ化した値をさらに RIPEMD-160 でハッシュ化し、その値を Base58 エンコードするという過程を経る。

いま、ユーザ X はユーザ Y に金額 M の送金を企図してトランザクション T を発行するとする。トランザクションの標準構造は複数あるが、ここでは執筆時点で最も利用されている P2PKH のものとする。まず、ユーザ X は金額Mを用意するために、インプットトランザクション内に過去に X 宛に送られた未使用のアウトプットトランザクションを指定し、X は自身の秘密鍵による電子署名と公開鍵を用いてスクリプト（ScriptSig）を記述する。同時に、アウトプットトランザクション内に金額 M、出力先として Y のアドレス、Y の公開鍵と Y の秘密鍵による電子署名の提示を解除条件とするスクリプト（ScriptPubKey）を記述する。インプットトランザクションとアウトプットトランザクションを主な構成要素としてトランザクション T が生成される。インプットトランザクションは送金者が送金金額を用意するもの、アウトプットトランザクションは相手の宛先と送金金額を示したもの、それら 2 つを主要な構成要素として価値の移転を記述したものがトランザクションであると理解するとよいだろう。インプットトランザクションのように過去のアウトプットトランザクションを参照して送金金額を用意する理由は、ビットコインは銀行のようにアカウントベースで残高を管理しておらず、ブロックチェーンに記録されている全トランザクションを辿って所持金額を算出する仕組みを取っているためである。トランザクション、インプットトランザクション、アウトプットトランザクションの構造を表 2.1 から表 2.3 に示す[3]

実際に送金を考えているのはユーザ X だけではないので、トランザクション T のようなトランザクションは多く存在する。ノードらは、トランザクション T を含む多数のトランザクションからブロック B を生成しようとする。トランザクションそれぞれについてインプットトランザクションの送金者の電子署名の検証等を行うことで有効性を確かめ、無効なトランザクションは除外する。このようにしてブロックサイズ上限の 1MB を超えない範囲でトランザクションを取り込んでいくが、ノードは自由に取り込むトランザクションを選択できるので、通常、ブロックはノードごとに異なる。ブロックの構造を表 2.4 と表 2.5 に示す[4]。

ノードらはブロックの生成権をめぐり、計算問題を解く競争を行う。具体的には、①ブロック B の親ブロックの持つブロックヘッダのハッシュ、②B に含まれる全トランザクションのマークルルートハッシュ、③ナンスという 3 つの値をもとにハッシュ化を行い、予め定められた値より小さなハッシュを探す。①、②はそれぞれ SHA-256 を 2 回使ってハッシュ化した 256bit の値である。③はその小さなハッシュを見つけるべくノードが 0 からインクリメントして試行するための値である。条件を満たすハッシュを最も早く見つけたノードにブロックを生成する権利が与えられる。勝利したノードはブロック B を生成し、ネットワークへブロードキャストする。その他のノードらは、ブロック B を受信したら B の①～③の値をもとに正しく計算問題を解いていることを確認して自らのブロックチェーンに取り込み、他のノードにも送信して、次のブロックの生成に取り掛かる。①のブロックヘッダのハッシュは分岐がない限り競争を行うノードで共通であるが、②はノードごとに取り込むトランザクションが異なるため同じマークルルートハッシュにはならないので、ノードごとに全く異なる計算をしなくてはならない点に留意されたい。このようにハッシュパワーに基づいてコンセンサスを形成するアルゴリズムを PoW（proof of work, プルーフオブワーク）という。ブロック B が作られてから次のブロックが作られるまでの時間、つまりブロックタイムが約 10 分となるように計算問題の難易度はネットワークによって自動的に調整される。

こうして正常にトランザクション T がブロック B に含まれると、ユーザ Y は自らの公開鍵と秘密鍵による電子署名を示すことでユーザ X が施したロックを解除して利用できるようになる。

以上、本セクションでは基本的なブロックチェーンとしてビットコインについて取り上げて構造を概説した。おおよそのブロックチェーンで上記説明の流れは概ね共通であると思われるが、具体的に見ていくとブロックやトランザクションの構造はもちろん、コンセンサスアルゴリズム、もっと基本的なプロトコルといったレベルで相違点がありブロックチェーンごとに様々な特色がある点を付け加えておく。

## 3.ブロックチェーンの課題と解決策

ブロックチェーンについて課題は複数あるが、取りうる解決策は基本的に、オンチェーン（Layer1）での解決か、オフチェーン（Layer2）での解決かのどちらかである。オンチェーンは、当該のブロックチェーン自体のプロトコルを変更するものである。当該ブロックチェーンを維持するネットワークの帯域幅、ノードのストレージ、メモリ等のリソースを勘案して行うことになる。オフチェーンは、当該のブロックチェーンで行う処理を切り離して

表 2.1 トランザクションの構造

| フィールド名 | 説明 | データサイズ |
|---|---|---|
| バージョンナンバー | バージョン | 4 バイト |
| フラグ | あれば 0001 で、ウィットネスがあることを示す | あれば 2 バイト、なければ 0 バイト |
| インプットカウンタ | インプットトランザクションの数 | 1-9 バイト |
| インプットリスト | インプットトランザクション（表 2.2 参照）のリスト | インプットトランザクションの数による |
| アウトプットカウンタ | アウトプットトランザクションの数 | 1-9 バイト |
| アウトプットのリスト | アウトプットトランザクション（表 2.3 参照）のリスト | アウトプットトランザクションの数による |
| ウィットネス | あれば 1 インプットにつき 1 ウィットネス | 可変 |
| ロックタイム | トランザクションのロックが解除されるブロック高かタイムスタンプ、ロックなしの場合は 0 | 4 バイト |

表 2.2 インプットトランザクションの構造

| フィールド名 | 説明 | データサイズ |
|---|---|---|
| 前トランザクションのハッシュ | このインプットが参照している前トランザクションのハッシュ | 32 バイト |
| 前アウトプットトランザクションのインデックス | このインプットが参照している前トランザクションのアウトプットトランザクションの番号(先頭のアウトプットトランザクションは 0) | 4 バイト |
| 入力者のスクリプト長 | 入力者の署名スクリプトの長さ | 1-9 バイト |
| 入力者の署名スクリプト（ScriptSig） | 入力者を確認するためのスクリプト、入力者の秘密鍵による電子署名と公開鍵を含む | 可変 |
| シーケンス | 送信者が定義するトランザクションのバージョン | 4 バイト |

表 2.3 アウトプットトランザクションの構造

| フィールド名 | 説明 | データサイズ |
|---|---|---|
| 値 | トランザクションの送金額 | 8 バイト |
| 出力先のスクリプト長 | 出力先のスクリプトの長さ | 1-9 バイト |
| 出力先のスクリプト（ScriptPubKey） | 出力先の公開鍵を用いて値を引き出すための条件を記述したスクリプト | 可変 |

表 2.4 ブロックの構造

| フィールド名 | 説明 | データサイズ |
|---|---|---|
| マジックナンバー | 0xD9B4BEF9 固定、ネットワークを識別 | 4 バイト |
| ブロックサイズ | ブロックヘッダ～トランザクションまでのブロックサイズ | 4 バイト |
| ブロックヘッダ | ブロックのヘッダ情報（表 2.5 参照） | 80 バイト |
| トランザクションカウンタ | ブロック内のトランザクション数 | 1-9 バイト |
| トランザクション | トランザクションのリスト | 可変 |

表 2.5 ブロックヘッダの構造

| フィールド名 | 説明 | データサイズ |
|---|---|---|
| バージョン | ブロックのバージョン | 4 バイト |
| 親ブロックのハッシュ | 親ブロックのブロックヘッダをハッシュ化したもの | 32 バイト |
| マークルルートハッシュ | ブロック内の全トランザクションをハッシュ化したもの | 32 バイト |
| タイム | ブロック生成時のタイムスタンプ | 4 バイト |
| ビット | ブロック生成時のプルーフオブワークの難易度 | 4 バイト |
| ナンス | 0 から始まる 32 ビットの数字 | 4 バイト |

表 3.1 スループットの比較

| 比較対象 | 平均スループット | 最大スループット | 備考 |
|---|---|---|---|
| ビットコイン | 2.6tps | 8.3tps | 平均スループットは[5]より、2017/10/22～2018/10/21 のデータについて平均を算出。最大スループットは、ブロックサイズ 1MB、平均トランザクションサイズ 200B、ブロックタイム 600 秒で試算。 |
| イーサリアム | 8.2ps | 20tps | 平均スループットは[6]より、2017/10/22～2018/10/21 のデータについて平均を算出。最大スループットは[7]による。※イーサリアムは PoW 型から PoW/PoS 併用型、あるいは PoS 型への移行を検討している点を付け加えておく。 |
| PayPal | 240tps | 不明 | 平均スループットは[8]による。 |
| VISA | 1736.1tps | 56000tps | 平均スループットは[9]をもとに算出。最大スループットは[10]による。 |

他のブロックチェーンに移すものである。オンチェーンのリソースの問題は無視できるが、技術自体を開発する必要がある。

また、厳密には課題の解決策とは言えないが、新しいブロックチェーンを開発することも選択肢の一つである。既存のブロックチェーンで実現不可であるか、実現にあたって犠牲にしたくない何らかのトレードオフがあるような機能、あるいは特定目的の達成に特化したものを開発したい場合には採用される場合がある。

以下では、ブロックチェーンにおける課題としてスケーラビリティを取り上げて説明し、解決策についても簡単に触れる。

### 3.1.スケーラビリティ

多くの初歩的なブロックチェーンで毎秒処理できるトランザクションの数は非常に少ない。これは、実用上の課題であるとされている。ユーザ数の増加、ユーザの利用の増加につれてこの問題は致命的になる。トランザクションの処理待ち時間は増え、ノードは演算能力を恒常的に稼働させなくてはならなくなる。代表的な PoW 型のブロックチェーンであるビットコインとイーサリアム、また取引の処理をしているシステムの例として PayPal や VISA を取り上げてスループット（transactions per second, tps）を比較した結果を表 3.1 に示す。ブロックチェーンは既存のシステムと比べてスループットが低いことが分かる。

### 3.2.オンチェーンでの対応

オンチェーンで解決を図る場合、最も単純な方法はブロックチェーンを定義するシステムパラメータを変更することである。例えばブロックサイズを 2 倍にすれば、平均的に 2 倍のトランザクションを含めることができるようになる。ブロックタイムを 1/2 にすれば、1/2 の時間で同数のトランザクションを処理できるようになる。しかし、この方法はネットワークの帯域幅、ノードのメモリ、ストレージ、演算能力等に負荷をかけることになるので、それらを考慮して決定しなければならない。また、この方法でスケーリングできる限界もある。

別の方法としてはコンセンサスアルゴリズムを変更する方法がある。ただし、コンセンサスアルゴリズムの変更は単純にスケーラビリティ問題だけを考えて行われるようなものではない。新しいコンセンサスアルゴリズムの採用によりブロックチェーン

の仕組み自体が変化するので、攻撃手法やその対策、ネットワークの設定等、多方面の検討と改修が必要となることが多い。むしろ、新しいコンセンサスアルゴリズムを採用する場合は、ゼロからブロックチェーンを開発する方が少ない工数で済むという考えもある。コンセンサスアルゴリズムの選択に関しては、第 3 章の 4 で補足する。

### 3.3. オフチェーンでの対応

オフチェーンで解決を図る場合、ステートチャネルやサイドチェーンを用いた技術が利用される。例えば、ユーザ X とユーザ Y の間で複数回の送金を行う場合を考える。この時、X と Y にとって最も重要な情報は双方の最終的な金額がいくらになるかであり、その中間の履歴は必ずしも必要でない。ステートチャネルやサイドチェーンを用いた技術は、この中間の履歴に当たる処理を代行することで、当該のブロックチェーンで発行されるトランザクション数を削減する技術である。トランザクションの数が減るということはそれだけストレージを節約できるということであり、インプットトランザクションやアウトプットトランザクションにあるスクリプトの署名検証作業に割いていたリソースも節約できるということである。これによりスケーリングがなされる。両技術についてもう少し詳しく見ていく。

ステートチャネルは、通常ブロックチェーン上で行われるステート（状態）の更新をブロックチェーン外で行おうとするものである。主に以下のステップで行われる。まず、マルチシグネチャ等でブロックチェーンのステートの一部をロックし、一定数の参加者の合意があれば更新できるようにする。次に、トランザクションを発行し参加者間でステートの更新を実行できるようにする。このトランザクションはブロックチェーンに送信できるものであるが、送信は保留しておく。新しい更新は古い更新に優先する。終わりに、参加者はブロックチェーンにステートを返してステートチャネルを閉じ、ブロックチェーンでロックしていたステートのロックを解除する。

この技術を送金用途で応用したものがペイメントチャネルである。ペイメントチャネルにおいてステートの遷移とは、送金の遷移である。ただし、やりとりが直接チャネルを開設している二者間のみに限定される、チャネルの開設に保証金（ビットコインであればビットコインのデポジット）がいる、受取人もオンラインである必要がある、トランザクションマリアビリティの問題がある、通常の送金に比べて遥かに仕組みが複雑であるといった課題がある。ただし、チャネルが二者間のみに限定される点に関しては、例えば HTLC（Hashed Timelock Contract）などの技術により解消されつつある。以上によりブロックチェーンで行われていた中間のトランザクションをブロックチェーン外で実行できるので、スケーラビリティが改善される。

サイドチェーンは、2way-peg と総称される手法を用いてブロックチェーンとトランザクションのやりとり等をできるようにする技術である。サイドチェーンは、ペッグされるブロックチェーンとは全く別のブロックチェーンである。例えばチェーンを維持するノードやコンセンサスアルゴリズム、システムパラメータまで全て異なっていてよく、ペッグされるブロックチェーンの設計に縛られないので開発の自由度が非常に高い。従って、今回はスケーラビリティの文脈でサイドチェーンについて述べているが、必ずしもそれだけに限らず利用される技術である。

スケーラビリティの観点から、サイドチェーンに用いられるコンセンサスアルゴリズムは PoS（Proof of Stake, プルーフオブステーク）をベースとする高速なものが多く、数千 tps を記録するものも出現している。PoW のようにノードのハッシュパワーによる演算速度に基づいてブロックの生成確率を高める代わりに、PoS はノードがロックしたステークに基づいてブロックの生成確率を高める。PoS はチェーンベースの PoS（chain-based PoS）と BFT スタイルの PoS（BFT-style PoS）に分かれる。

チェーンベースの PoS は、アルゴリズムが擬似的にランダムにノードを選び、選ばれたノードがブロックを生成できるというものである。生成されるブロックは、直前のブロックを参照している必要がある。BFT スタイルの PoS は、ランダムに選んだノードにブロックを「提案」する権利を与えて、どのブロックを正とするかについてノードが「投票」することで合意を形成するようなプロトコルである。合意が形成されると確実なファイナリティが与えられて、覆ることはなくなる。チェーンベースと比べると、1 ブロックに対して合意が形成され、なおかつ確実なファイナリティが与えられるなどの違いがある。どのような実装をするかは要求される可用性や一貫性を考えて決定することになる。

### 3.4.コンセンサスアルゴリズムの選択にみるトレードオフ

スケーリングをするにあたり、高速なコンセンサスアルゴリズムを用いればよいと考えるのは誤りではない。ただし、高速コンセンサスアルゴリズムの採用にはトレードオフがある点を押さえておきたい。Vlad Zamfir は次のような図 3.1 を用いてトレードオフをうまく表現している[11]。

ファイナリティの速度（＝ファイナリティが速いか、遅いか）、

オーバヘッドの多寡（＝ネットワークの余分な仕事量が多いか、少ないか）、ノードの数（＝数が多く非中央集権的か、数が少なく中央集権的か）のうち、選択できるのは 2 つだけである、というトリレンマである。

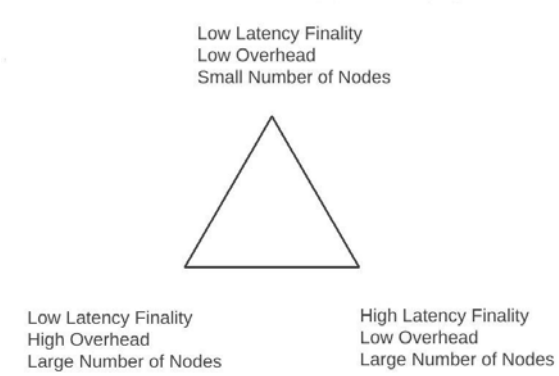


図 3.1 Vlad のトレードオフ

PoW 型（例：ビットコイン）のブロックチェーンを考えてみる。ブロックチェーンの仕組みによればコンセンサスが形成されてチェーンが長くなる度、過去のブロックが改竄される可能性は指数関数的に下がっていく。ただし 0 にはならないので、ファイナリティはあくまで確率的なものである。ブロックタイムも 10 分と長い。しかしながらノードが行っていることはプロトコルに定められたものだけであり、多数のノードがブロックの生成競争をすることに鑑みて、コンセンサスに参加するノード数も非常に多いと言える。Vlad の三角形の右端に位置すると言えるだろう。

一方で、100%のファイナリティをもたらすプロトコルの例として PBFT（practical byzantine fault tolerance）を考えてみる。PBFT は、ノードが互いに通信しあい、2/3 以上のコンセンサスを得た場合にファイナリティを定めるというものである。ファイナリティは即時であることから、この点に関しては優れていると言える。ただし、非中央集権性とスケーラビリティの間にはトレードオフがある。ノードが少なければ中央集権的になるし、攻撃可能性も高まる。一方でノードを増やせば通信回数が増える。単純にノードが n 個増加すると通信は二次関数的に増加する。スケーリングが容易でない環境下ではノードを限定せざるを得ないが、限定するとブロックチェーンの非中央集権性という特徴が失われがちになる。さらに、ノードを予め認識しておく必要があるので、自由にオフラインになることができないという制約もある。Vlad の三角形の左端と上端を結ぶ辺で設計を模索することになる。

また、1/2 や 1/3 等のコンセンサスが正常に形成されるための閾値や中央集権的か非中央集権的かといった違いはセキュリティにも影響する。例えば、PoW では悪意ある攻撃者が 1/2 を超えるハッシュパワーを保有するならば遡ってチェーンの改竄が可能になる。チェーンベース PoS であれば、ハッシュパワーの代わりに過半数のステークで改竄可能である。

これに対し、BFT スタイルをとる PBFT では 1/3 より大きいノードを掌握すれば、過去に遡っての改竄はできないがコンセンサスを止めることができる。仮に悪意ある攻撃者の数が 0 だとしてもノードの 1/3 以上がオフラインであればコンセンサス形成自体が止まる。一方、PoW は 1/3 のノードが失われても正常にコンセンサスは形成される。また、コンセンサス形成に関わるノードが少なければ、コンセンサスを止めるのに必要なノードの数も当然少なくなる。コンセンサス形成にかかわるノードの数が十分に少なければ、そのノードをお金で懐柔するような攻撃も考えられる。PoW 型の、例えばビットコインのようなブロックチェーンで過半数のハッシュパワーに達するまでノードをお金で懐柔しようとするのは無謀であろう。

このように、コンセンサスアルゴリズムの選択はトレードオフがあるだけでなく、攻撃の手口や攻撃のインセンティブが変わってくるなど、セキュリティにも影響を及ぼすことから、慎重さが求められるだろう。

## 4.展望

ブロックチェーンのもたらす社会的な影響について論じるにあたり、1 つの観点として機能面に焦点を当ててブロックチェーンの特徴をまとめる。なお、メリットとなるかデメリットとなるかは文脈に依存するため、そのような観点での整理は行わない。

1. 原則として記録のテクノロジであるが、物理的に何かを生成するような技術ではない。

   ブロックチェーンは基本的には何らかのデータ、並びにデータの変化を記録する技術と言える。従って、例えば車や PC のように物理的な何かを製造する技術を代替することはできない。

2. 電子データの生成やロジックの記述ができるが、サイズの大きなデータや重たいロジックの扱いは現実的でない。

   ブロックチェーンはネットワークを構成するノード（フルノード）にブロックチェーンそのものを持たせる。また、スマートコントラクトを利用して電子的なデータ（トークン等）を生成することや、生成したデータの処理ロジックを記述することができる。ただし、ノード側のストレージ

の問題から、大量のデータを書き込むことは難しい。また、重たいロジックの実行も処理を担当するノードが十分な演算能力を有するとは限らないことから現実的でない。

3. 透明性と追跡性、擬似的な匿名性があるが、現実のアイデンティティと紐づくとプライバシーの問題がある。

   ブロックチェーンのトランザクションは全て記録され、誰でも閲覧できる状態に置かれるので不正がないことを示すことができる。また、アドレスによる擬似的な匿名性が確保されている。しかし、ひと度現実のアイデンティティと紐づくとブロックチェーンの解析によりかなりのプライバシー情報が推測されてしまう。

4. 耐障害性があるが、設計に注意が必要である。

   複数のノードが同一のブロックチェーンを持っていることで、ネットワークの一部に障害が発生しても総体として正常に機能する。ただし、ノードのふるまいやネットワーク分裂時の対応などに関して繊細な設計を要する。

5. 入力を正しいものとして処理するが、入力自体の正しさを保証するわけではない。

   ブロックチェーンは入力を正しいものとして処理し、記録する。コンセンサスアルゴリズムにより、ブロックチェーン内で正しく処理されたことは確率的に保証される。また、確実なファイナリティを有するコンセンサスアルゴリズムも登場している。しかし、その入力自体が正しいものかどうかについて保証しない。例えば、ユーザ X が署名付きトランザクションを Y に対して送信したとしても、このトランザクションの現実の送信者が X であることを保証しないし、受信者が Y であることも保証しない。

6. 改竄耐性を持つが、ロールバックができない。

   コンセンサスアルゴリズムに基づいてファイナライズされた値に対して高い改竄耐性を持つが、トランザクションの内容に誤りがあるなどして後から変更を加えたい場合でもブロックに取り込まれると元に戻すことができない。

7. 非中央集権により検閲耐性を得るが、多くのステークホルダーが絡む。

   ブロックチェーンは中央で管理する主体を持たず、プロトコルに基づいてネットワークが維持する。その代わりに、プロトコルの更新等を行おうとすれば開発者だけでなくネットワークに参加しているノードの合意を得る必要があるなど、多数のステークホルダーが関係するため改修が容易でない側面がある。

1 は前提である。2〜7 は特徴とそのトレードオフの例を書いたもので、ブロックチェーンの選択を考慮するうえで検討すべき観点となるだろう。また、2〜7 については、第 3 章に示したものも含めて技術の選択と実装によってある程度自由に性質を強めたり弱めたりすることができるところまで来ていると考えてよい。以上の点から考えて、ブロックチェーンが既存のデータベース等の記録のテクノロジを駆逐するわけではないが、目的に応じてブロックチェーンに置き換えられることは十分に考えられる。いまは決済関連のサービスが多いが、ブロックチェーンだからこその有効性を発揮するようなサービスも出てくるようになるだろう。

ブロックチェーンが適していると考えられる分野として権利関係の取扱いが挙げられる。一例として、ゲーム市場を見てみる。ゲームは、ユーザのアカウントがあり、アカウントにステータスなどのパラメータが設定されていて、アイテムなどのアセットがあり、そのアセットにも所有・消費・失効などの概念がある場合が多い。一部のアセットは法定通貨で、または法定通貨をゲーム内通貨に変換して購入（いわゆる課金）することができる。また、日本国内では禁止しているところが多いが、ゲームによってはアセット to アセット、あるいはアセット to 法定通貨のユーザ間トレードが行われている場合もある。

このようなゲームのアプリケーションロジックやアセットの所有権、マネーとの交換による所有権の移転などのイベントはスマートコントラクトで記述し、ブロックチェーンに記録できる。スマートコントラクトを記述することは、ブロックチェーンでプログラミングを行うことと考えると分かりやすい。

ブロックチェーンを用いることで得られるメリットとして、例えばバックアップコストがかからなくなること、サーバメンテナンスが不要になること、障害によるサービス停止リスクが少なくなること、ユーザの個人情報やパスワード等のセンシティブデータを保持せずに済むことなどが考えられる。過去、あるオンラインゲームがサーバの不具合からメンテナンスを実施し、データ復旧ができずにサービス終了となった事例[12]があるが、ブロックチェーンでそのようなことは考えにくい。ブロックチェーンの維持については、ブロックチェーンのコインをゲーム内で利用できるリアルマネーと同等の通貨としたうえで、リソースを貸し出してくれるユーザにコインを報酬として付与する等のエコシステムを設計することなどが考えられるだろう。一方、アクティブなユーザが多くなれば当然トランザクションが多数発生しブロックチェーンの維持ノードに負荷をかけるが、これはスケーラビリ

ティ問題とその解決策で述べたように対策が考案されつつある。

アセットや権利、それらの移転などの定義や記述はゲームだけに留まらない。例えば、証券やポイントなど様々なものを置き換えていくトークナイゼーション（tokenization）の動きも起こっている。

また、ブロックチェーンはマイクロペイメントの可能性を拓く。マイクロペイメントは、法定通貨では通常不可能な 1 円単位よりも小さな単位での決済を信頼できる第三者機関なしに二者間で直接的に行うものである。さらに、ペイメントチャネルやサイドチェーンといった技術を組み合わせると、超小額決済を高速かつ不可逆的かつ不正なしに完結できるようになる。例えば、サービスを「特定の時間で何らかの価値を提供するもの」と簡単に捉えてみると、サービスに対する支払いは時間に対する支払いと価値（コンテンツ）に対する支払いとに分類できる。すなわち、マイクロペイメントが可能になることは時間やコンテンツを極めて細かく分割する力を個人に与えることを意味する。これにより、動画の閲覧秒数に対する課金や、有料記事をさらに細分化して関心のある部分のみ閲覧したい読者に対して文字数ベースで課金するなどの新たなモデルが誕生する可能性がある。

当然、ブロックチェーンのみで完結できるとは限らない。先の動画の例であれば、動画自体をブロックチェーンに記録することはノードのストレージの問題から不適切である。ゆえにストレージサービスに動画自体を置き、オーソリティをブロックチェーンで管理するような構造になるかもしれない。設計に関しては多様な可能性がありそうだが、いずれにせよマイクロペイメントは新分野を開拓する可能性を秘めていると考えられる。

権利関連やマイクロペイメントを取り上げて見てきたが、大切なことは、ロジックやアセットを電子的に定義でき、ブロックチェーンに不可逆的に記録し、状態遷移を明確に追跡できるという性質が普遍的なものだということであり、それぞれ特性の異なる多様なブロックチェーンが誕生し、文脈によって選択できる可能性が生まれているということである。

## 5.終わりに

本稿では、まずビットコインを取り上げて基本的な構造を概観した。そのうえで、多くのブロックチェーンが課題として直面しているスケーラビリティ問題を取り上げ、オンチェーンとオフチェーンによるスケーリングについて簡単に紹介した。特に、高速コンセンサスアルゴリズムを用いたサイドチェーンは研究や開発が盛んに行われている。中央集権性等とのトレードオフは存在するものの数千 tps ほどのスループットを持つブロックチェーンも登場してきており、クライアントサーバ型の決済システムと比較しても劣らないほどの実用性が備わりつつあるように思う。また、展望ではブロックチェーンの特徴を整理し、ロジックの記述、ブロックチェーンへの恒久的な記録、状態遷移の追跡といった普遍的な性質により、多くの分野への応用や新天地の開拓ができるポテンシャルがあることを述べた。

一方で、本稿で取り上げたスケーラビリティ以外の課題も山積しているということは付け加えておきたい。また、ブロックチェーンを利用することがかえってマイナスに作用する分野も多くあると思われる。

ブロックチェーンの応用は緒に就いたばかりである。いま利用されている多くの技術がそうであったように、応用しては問題点を指摘され、問題点の解消方法を開発してはまた応用するというような改善のプロセスを通して、その真価を十分に発揮するブロックチェーンが社会により良い影響をもたらすことを切に願う。

文献


[1]Satoshi Nakamoto, https://bitcoin.org/bitcoin.pdf, 2008

[2]日本ブロックチェーン協会, http://jba-web.jp/archives/20110003blockchain_definition, 2016

[3]Bitcoin Wiki, https://en.bitcoin.it/wiki/Transaction, 2018

[4]Bitcoin Wiki, https://en.bitcoin.it/wiki/Block, 2018

[5]Blockchain Luxembourg S.A., https://www.blockchain.com/ja/charts/n-transactions-per-block, 2018

[6]Etherscan, https://etherscan.io/chart/tx, 2018

[7]ETHNews, Jim Manning, https://www.ethnews.com/the-raiden-network-could-allow-instant-transactions-in-ethereum, 2016

[8]PayPal, https://www.paypal.com/stories/us/paypal-reports-fourth-quarter-and-full-year-2017-results, 2018

[9]VISA, https://usa.visa.com/run-your-business/small-business-tools/retail.html, 2018

[10]VISA, https://usa.visa.com/dam/VCOM/download/corporate/media/visa-fact-sheet-Jun2015.pdf, 2015

[11]Vlad Zamfir, https://twitter.com/VladZamfir/status/942271978798534657, 2017

[12]ハンゲーム, http://info.hangame.co.jp/index.nhn?m=detail&infono=4385&tab=ALL, 2011