ブロックチェーンアルゴリズムの分類と問題点
Classification and issue of “Block chain algorithm”

荒 牧 裕 一
ARAMAKI, Yuichi

はじめに

ブロックチェーンは、ビットコインを始めとする仮想通貨における分散型台帳技術として開発されたものであるが、これを仮想通貨以外の用途で活用しようという動きがあり、「ブロックチェーン2.0」と呼ばれている。

しかし、ブロックチェーン2.0においては、ビットコインにおけるブロックチェーンとは異なる性質を持つものも多く、一口にブロックチェーンといってもその技術的特徴は様々である。

本論文では、現時点でのブロックチェーンの技術的特徴と問題点について整理していく。まず、分散型台帳システムとしてのブロックチェーンの一般的な定義を行う。そして、様々なブロックチェーンについて、ノードの公開性および承認アルゴリズムの観点からの分類を紹介する。最後に、これら分類を踏まえて、ブロックチェーンの有する問題点を整理する。

Ⅰ．ブロックチェーンとその定義

1．ブロックチェーンとは

ブロックチェーンとは、2009年にSatoshi Nakamoto氏の論文[1)]で提唱された仮想通貨（暗号通貨とも呼ばれる）であるビットコインの根幹技術として生まれた分散型台帳システムである。本来は仮想通貨の根幹技術として実用化されたが、その後その記録の改ざん困難性等の特徴が注目され、ブロックチェーン2.0（または、ビットコイン2.0）と呼ばれる仮想通貨以外の用途での活用が進められている。

2．ブロックチェーンの定義

Satoshi Nakamoto氏の論文中では「ブロックチェーン」という言葉自体は用いられておらず、その定義も明確ではない。そこでまず、ビットコインおよびブロックチェーン2.0で用いられている分散型台帳システムの技術的特徴を踏まえ、その定義を行いたい。

ビットコインにおけるブロックチェーンの技術的特徴は次のとおりである。このほかにゼロダウンタイムや安全性が挙げられることもある[2)]が、それらは技術的特徴ではなくビットコインの運用実績としての特徴であると考えられる。

① 仮想通貨の発行・配布機能を有する。(仮想通貨発行機能)
② P2Pベースの分散型ノード間でトランザクションデータを共有する。(P2P分散型システム)
③ P2Pはインターネット上で開放され誰でもノードとして参加できる。また、ソースコードはオープンソース化されている。(公開性)

④ 複数のトランザクションデータが１ブロックにまとめられ、台帳（以下、「チェーン」という。）に追加される。（ブロック生成）
⑤ チェーン内の各ブロックは、ハッシュ関数等[3]で繋がれ、新規ブロックの追加のみ可能（Append-only）であり、チェーン内の過去のデータを部分的に変更することは出来ない。（不可逆性）
⑥ チェーンへのブロック追加に当たり、参加ノードによる承認手続が必要となる。（承認アルゴリズム）

このうち、①の仮想通貨発行機能はビットコイン等の仮想通貨システムでは当然必要であるが、仮想通貨以外の用途での活用を目指すブロックチェーン2.0では必ずしも必要ではない。また、③の公開性についても、仮想通貨システムのような不特定多数を対象にしたシステムでは必要であるが、特定の業界や企業内のみで利用されるシステムでは不要である。よって、ブロックチェーンの一般的な定義としては、上記①および③を除いた４要件を満たした台帳システムを指すものであると考えられる。（図１参照）。

ブロックチェーンとは、以下の４要件を備えた台帳システムである。
(1) P2P ベースの分散型ノード間でトランザクションデータを共有する。（P2P 分散型システム）
(2) 複数のトランザクションデータが１ブロックにまとめられ、チェーンに追加される。（ブロック生成）
(3) チェーン内の各ブロックは、ハッシュ関数等で繋がれ、新規ブロックの追加のみ可能（Append-only）であり、チェーン内の過去のデータを部分的に変更することは出来ない。（不可逆性）
(4) チェーンへのブロック追加に当たり、参加ノードによる承認手続が必要となる。（承認アルゴリズム）

図１　ブロックチェーンの定義

Ⅱ．ブロックチェーンの分類

１．ノードの公開性における分類

ブロックチェーンの分類においては、まず、参加ノードの公開性に応じた３分類が挙げられる。[4]

(1) パブリック（Public）
不特定多数の者が P2P ネットワークへ参加できる。
悪意を持った参加者が加わるリスクがあるため、そのリスクを排除できる厳格なブロック承認手続が必要である。

(2) コンソーシアム（Consortium）
特定の団体や企業グループに属する者のみが P2P ネットワークへ参加できる。
事前に参加者の身元確認が取れているため、比較的緩やかなブロック承認手続で運用できる。

(3) プライベート（Private）
単一の組織や企業内の者のみが P2P ネットワークへ参加できる。

参加者の身元確認が取れているほか、管理者が一元化されているため、高速処理や頻繁なバージョンアップによる性能向上が円滑に出来る。

管理者が一元化できる反面、分散型システムの利点が薄らぐ。

2．承認アルゴリズムによる分類

ビットコインの登場後、新たに派生した仮想通貨やブロックチェーン2.0においては、ブロック承認手続として以下のようなアルゴリズムが採用されている。[5)]

(1) Proof of Work（PoW）

参加者が、特定の条件を満たす解を求める計算を繰り返し、最初に解を求めた者にブロック追加の権利を与える方法である。

ビットコインでは、単純なハッシュ関数による計算が採用されているため解の難易度が非常に高まり、資源の浪費が問題となっている。そのため、Litecoin 等の派生仮想通貨の多くは Scrypt（S-Crypt）による計算方法を採用して難易度の上昇を抑えている。

PoW の問題点としては、上記の資源浪費問題の他、総計算能力の過半数を占めるノードであれば虚偽のブロックをメインチェーンとして作成・維持することが可能となる危険性（51%攻撃）が指摘されている。

(2) Proof of Stake（PoS）

仮想通貨の保有割合や保有期間等に応じて計算されたコイン年数[6)]に基づき、ブロック追加の優先権を与える方法である。PoS でも、ブロックの追加のためにはハッシュ関数等の計算による解が必要となるが、コイン年数が大きいノードに対しては、より難易度の低い計算問題を割り当てることで優先権を与えている。

悪意を持った者が短期間に大きなコイン年数を獲得することが難しいため、単純な51%攻撃は防止できる。しかし、資金の豊富な攻撃者が仮想通貨の過半数を獲得する意思とそのための十分な資金の証拠を表明した場合、仮想通貨の価格が暴落するため、低コストでの51%攻撃が可能になるという危険性が指摘されている[7)]。

また、分岐（フォーク）が生じても、直前のコイン年数は分岐した双方のチェーンで同等に機能するため、どちらか一方のチェーンをメインチェーンとして維持するという動機が生じないという「何も賭けていない（nothing at stake)」問題も指摘されている[8)]。

(3) Proof of Importance（PoI）

ノードごとの取引額・残高を指標とした取引グラフ分析により、残高と取引状況をクラスタリングして、個別のノードの重要性を計算し、より重要性の大きなノードに優先権を与える方法である。

PoS の改良型であるといえる。

(4) Practical Byzantine Fault Tolerance（PBFT）

「ビザンチン将軍問題（後述Ⅲ(5)②)」を解決するためのアルゴリズムである。

ノードの総数が既知で、不正ノード数の上限が決められるなどの条件が必要であり、パブリックなシステムへの適用は難しいとされる。

III. ブロックチェーンの有する問題点

(1) フォーク（分岐）が可能である

ブロックチェーンには不可逆性があるため、過去のチェーン内のデータを部分的に変更することは出来ない。したがって、同一系統のチェーンのみを見れば、過去のデータの改ざんは事実上不可能とされる。しかしこれは、全体で一つの系統のチェーンが維持された場合の性質に過ぎない。ブロックチェーンは、あるブロックに複数のブロックを繋げることによって分岐（フォーク）させることが可能である。分岐の原因は、複数ノードがほぼ同時に解を見つけるといった偶発的な場合もあるし、互換性の無いバージョンアップにより新旧のソフトウェアが一時的に混在したり、悪意を持ったノードが虚偽の取引データを含むブロックを繋げるといった意図的な場合もある。ブロックチェーンの構造上、この分岐の発生自体を防止することは出来ない。

ビットコインを始めPoWのブロックチェーンにおいては、分岐が生じた場合は、それが偶発的か意図的かに関わらず、最もブロック長が長い（難易度の累計が大きな）チェーンをメインチェーンと見なすルールを採用している。チェーンの維持に大量の計算能力を必要とするPoWでは、どれか一つのチェーンの維持に計算能力を集中させることが合理的であり、悪意を持ったノードが意図的に分岐をさせたとしても、他の多くの（少なくとも総計算能力の過半数を有する）ノードが正当なチェーンの維持に計算能力を投下し続ける限り、正当なチェーンがメインチェーンとして維持されることになる。そのため、参加ノード数が多く難易度も高いPoWのブロックチェーンでは、この問題は現実にはそれほど問題にならない。

しかし、PoSにおいては、上述した「何も賭けていない（nothing at stake)」問題が存在するため、分岐は大きな問題である。そのため、PoI等では、どれか一つのチェーンだけを維持するように重要性の要素を設定するなどして改良を図っている。

さらに、PoWでも、意図的に複数のノードを維持し続けることは可能である。先に、分岐が生じた場合は、最も長い（難易度の累計が大きな）チェーンをメインチェーンと見なすルールがあると述べたが、メインチェーンとならなかったチェーン（サブチェーン）は消えてなくなるわけではなく、参加ノードから無視されるだけである。したがって、意図的に分岐させたチェーンを一部のノードだけで維持し続け、サブチェーンとして存続させ続けることも可能である（図2参照)。事実、Ethereumのブロックチェーンでは、「The　Dao事件[9]」をきっかけに、メインチェーンのEthereumとEthereum Classicの2つのチェーンが並存している状況が続いている。

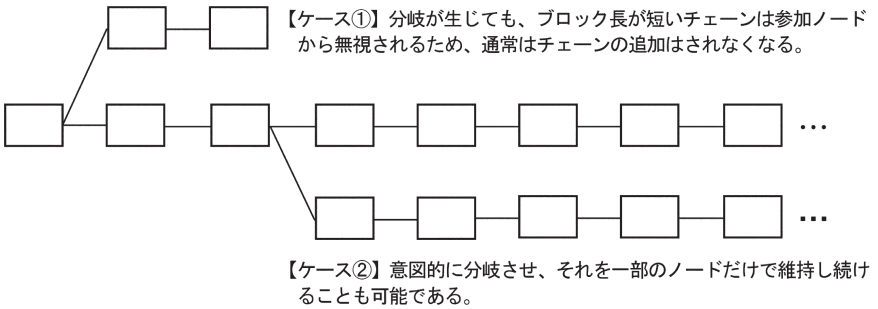

図2　ブロックチェーンの分岐（フォーク）

このように、複数系統のチェーンが同時に存在する可能性が高いブロックチェーンにおいては、二重譲渡や二重台帳の問題が生じるため、どれがメインチェーンであるかを客観的に確認できる仕組みが必要である。

これについて仮想通貨 Orb では、「スーパー・ピア」と呼ばれるノードがチェックポイント・ブロックを挿入してメインチェーンを確定する仕組みが採用されている。しかし、この仕組みは１台のスーパー・ピアに確定の権限を集中させるもので、P2P 分散型システムの利点を減殺させることになる。その他には、独立した第三者がノードとして承認作業に参加する仕組み等が考えられるであろう。

(2) 複数のブロックチェーンの併合は難しい

ブロックチェーンの分岐は容易である反面、既存のブロックチェーン同士の併合は難しい。これは、ブロックチェーンのデータベースが新規の取引だけを追加していく形式（Append-only）を採用し、過去のブロックを参照しないと最新の勘定残高がわからないことによる。

そのため、併合の必要がある場合は、新たなブロックチェーンを作成する形をとるものと思われるが、その際には、データの残高の確定、移行すべきデータの完全性・網羅性・正確性等について、十分な作業準備が必要となるであろう。

(3) 取引の確定（ファイナライズ）に時間がかかる

取引の確定とは、取引データを含むブロックの属するチェーンがメインチェーンとして確定することである。(1)で述べたとおりブロックチェーンには常に分岐のリスクがあり、しかも過去のブロックに遡って分岐させることも不可能でないことから、理論的には100％の確率で取引が確定することはあり得ない。

この問題について、ビットコインでは概ね６個のブロックが承認されれば取引が確定されたものと事実上見なしている[10]。これは、偶発的な分岐が生じた場合のサブチェーンのブロック数が通常は１～４であるという統計[11]に基づいている。しかし、この数値は、参加ノード数（多いほどリスクが高い）や承認作業時間（短いほどリスクが高い）によってリスクが変化するものであり、他のブロックチェーンでもそのまま採用できる基準ではない。

また、特に PoS では、最初期のブロックにおいて100％近いコイン年数を有しているノードがあれば、そのノードはその時点まで遡れば自由にチェーンを分岐できるのも大きな問題である[12]。そのため、PoS やその改良型のアルゴリズムでは、(1)で述べたチェックポイント・ブロック等の仕組みの導入が必要となる。

(4) 全ての取引がブロックチェーンに記録される保証がない

ビットコインではブロック内に記録する取引をデータ量と送金手数料を基準にして各ノードが選択するため、送金手数料を低く設定した取引は承認が後回しにされ、場合によってはタイムアウトで再度やり直さなければいけないことがある。この場合、ブロックチェーンに記録される取引の順序や網羅性について問題が生じることになる。

これについては、コンソーシアムやプライベートのブロックチェーンであれば、取引時間が早いデータを優先的にブロックに入れる仕組みを構築することも可能であろう。

(5) 分散処理システム一般の問題点

その他、分散処理システム一般において指摘されるものとして、以下の問題点も存在する。

①厳密な時刻管理が難しい

分散処理システムにおいては、基本的に処理を行ったノードが時刻を記録するが、全てのノードの時刻を厳密に統一することは、特にパブリック型の分散システムでは難しい。そのため、１秒間に多数の高速取引を行うシステムや、取引の前後関係を時刻によって厳密に判断すべきシステムには向いていない。

コンソーシアムやプライベート型のシステムなであれば、NTP あるいは独自プロトコルを用いて一定の時刻管理も可能な場合もある。

②ビザンチン将軍問題

分散処理システム上におけるノードのいずれかが偽の情報を伝達する場合に、全体として正しい合意を形成できるかという問題である[13]。

ビットコイン（PoW）では、先に述べた51％攻撃の問題に吸収されるため、事実上問題とはならないが、PoS や PoI では問題が顕在化する危険性がある。

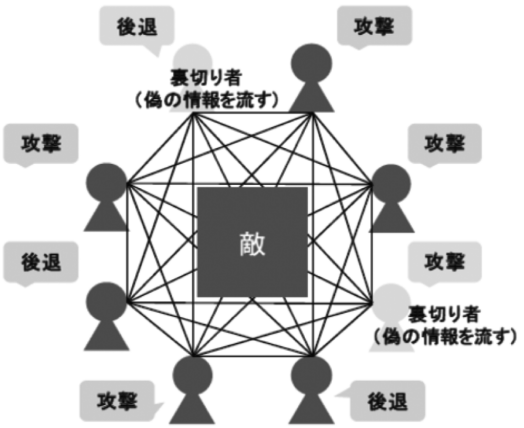

図 3　ビザンチン将軍問題[14]

③ＣＡＰ定理

分散処理システムにおいては、次の３つのうち完全に満たせるものは２つのみであるとされる問題である。

・C（Consistency、一貫性）：全てのノードで最新のデータを同時に保持している
・A（Availability、可用性）：特定のノードの障害により、他のノードが影響を受けない
・P（Partition-tolerance、分断耐性）：ネットワークに障害があっても継続して動作すること

ビットコインにおいては、A，Pを満たしているが、Cが犠牲になっていると指摘される。

おわりに

ブロックチェーンの技術的特徴と問題点について整理した。第一にブロックチェーンの一般的な定義を行い、「ブロックチェーンとは、① P2P 分散型システム、②ブロック生成、③不可逆性、④承認アルゴリズム、の４要件を備えた台帳システムである。」とした。第二に、様々なブロックチェーンについて、ノードの公開性および承認アルゴリズムの観点からの分類を紹介した。

第三に、ブロックチェーンの有する問題点として、①フォーク（分岐）が可能である、②複数のブロックチェーンの併合は難しい、③取引の確定（ファイナライズ）に時間がかかる、④全ての取引がブロックチェーンに記録される保証がない、⑤分散処理システム一般の問題点、について指摘をした。特に①の分岐の問題はブロックチェーンの本質的な問題点であり、どれがメインチェーンであるかを客観的に確認できる仕組みが必要であると考えられる。

以上

註

1）Satoshi Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System”, 2009年
2）ビットバンク株式会社、『ブロックチェーンの衝撃』編集委員会、『ブロックチェーンの衝撃』、2016年 6 月13日、p2，pp125-126
3）Litecoin 等では、ハッシュ関数ではなく Scrypt（S-Crypt）と呼ばれる暗号化アルゴリズムの一種が用いられている。
4）野村総合研究所、「平成27年度　我が国経済社会の情報化・サービス化に係る基盤整備（ブロックチェーン技術を利用したサービスに関する国内外動向調査）報告書」、2016年 3 月（2016年 4 月28日 経済産業省 HP にて公表）、p27
5）上掲注 4 ）、 pp9-10，p26
6）Sunny King, Scott Nadal, “PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake”, 2012年 8 月19日
7）上掲注 2 ）、p169
8）上掲注 2 ）、p168
9）代表的なブロックチェーン2.0でありスマートコントラクトを可能にするプラットフォームとしてリリースされた Ethereum（ETH）上のアプリケーション The DAO（投資ファンド）で生じた問題である。The DAO は、短期間で7,620,000ETH（当時の相場で約150億円）をクラウドファンディングで調達したが、そのコードに脆弱性があり、2016年 6 月17日に3,641,694ETH が流出した。ただし、仕様により流出した ETH が27日間保留されたため、その間に関係者が検討の結果、流出分を返還させるハードフォーク（互換性の無いバージョンアップによる分岐）を実施した。ハードフォーク自体は 7 月20日に無事実施されたが、ハードフォーク反対派が従来のチェーンを Ethereum Classic（ETC）と名づけて維持し続けて現在に至っている。
10）ブロックの承認によって新たに発行されたビットコインについては、100ブロックの承認を経なければ使用できないものとされる。
11）BLOCK CHAIN info の統計（https://blockchain.info/ja/charts/）によれば、2016年 8 月末までにサブチェーンのブロック数が 5 以上になったケースは11回（内 1 回は 7 ）あるが、全体のブロック数（42万強）に比べるとわずかであると言える。
12）上掲注 2 ）、p168
13）Leslie Lamport, Robert Shostak, Marshall Pease, “The Byzantine Generals Problem”，ACM Transactions on Programming Languages and Systems (TOPLAS) Volume 4 Issue 3，1982年 7 月，pp382-401
14）上掲注 4 ）、p12