修士論文

IoT 環境における分散台帳技術の性能分析

指導教官 林　冬惠

京都大学大学院情報学研究科
社会情報学専攻

渡辺隆弘

2019 年 1 月 19 日

IoT 環境における分散台帳技術の性能分析

渡辺隆弘

内容梗概

近年，「いつでも，どこでも，何でも，誰でも」ネットワークに繋がる「ユビキタスネットワーク社会」が構想されてきた．様々なデバイスがネットワークに接続されるようになると，それらのデバイス間での情報交換やデータの収集，それに基づく自動化が行われ，新たな付加価値を生むようになる．これらは幅広い分野での適用が検討されているが，収集，処理するデータの増加に伴い，深刻なセキュリティ，プライバシーの問題を引き起こすことが指摘されている．この問題を解決するために，分散台帳技術を採用することが検討されている．分散台帳技術は，取引データをネットワークに分散させ管理することで障害耐性，セキュリティを確保する技術であり，仮想通貨を始めとしてエネルギー分野，ヘルスケア分野などにおいてその可能性が期待されている．一方，既存の分散台帳技術をそのまま IoT 環境に転用することは困難である．これらは IoT 環境が，現在分散台帳技術が利用されている分野とは異なる特性を持つ環境であるためであり，IoT 環境に用いられるシステムは，スケーラビリティ，障害耐性などが特に求められる．本研究で取り組む課題は以下の 2 点である．

1. IoT 環境を考慮した分散台帳技術の評価指標の提示
   IoT 環境を想定した分散台帳技術について研究が行われている．このような技術開発において性能分析を行うための体系的な分析指標を提案する必要がある．その分析指標は，IoT という環境を考慮した指標である必要がある．

2. IoT 環境における分散台帳技術の性能評価手法の提案と分析
   分散台帳技術のアーキテクチャやアルゴリズムは様々なものが存在する．様々な技術特徴を持つ分散台帳技術を評価するための性能評価手法が必要となる．また，それらの手法に基づき性能を評価し，IoT 環境を踏まえた分散台帳技術の技術特徴について分析することが必要である．

本研究は，既存の種々の分散台帳技術を IoT 環境に適用する際，どのような評価指標を用いることが適切であるか議論を行い，その指標に基づき複数の分散台帳技術の性能比較を行う．そのため，IoT 環境を考慮し，分散台帳技術の評価基準のうち性能効率性とスケーラビリティに重点を置いた指標を作成する．また，トランザクションの伝播速度と承認速度を測定するためのシミュレータを作成し，性能効率性とスケーラビリティをシミュレーションにより測定し，性能の比較を行う．最後に，性能比較に基づき，IoT 分野に適用する際に適切であると考えられる分散台帳技術の技術特徴の分析を行う．
本研究の貢献は以下の 2 点である．

1. IoT 環境を考慮した分散台帳技術の評価指標の提示
   IoT 環境の特性を考慮した上で分散台帳技術を評価する評価指標を提案した．提案する指標は性能効率性，スケーラビリティを重点においた指標である．指標は，構造的特徴を捉える一般的な性質と，シミュレーションにより測定する定量的な指標からなる．

2. IoT 環境における分散台帳技術の性能評価手法の提案と分析
   提案した評価指標に基づき，Bitcoin の Blockchain，Ethereum，IOTA の 3 つの分散台帳について性能比較を行った．これらの分散台帳は異なる特徴を有しており，一般的な性質の比較と，定量的な指標についてのシミュレーションを行った．これらの性能比較を通じて，分散台帳技術を IoT 分野に適用する際に適切である技術特徴について考察し，提示した．

IoT 環境における分散台帳技術の性能分析

渡辺隆弘

内容梗概

a

目次

第 1 章 はじめに

近年，「いつでも，どこでも，何でも，誰でも」ネットワークに繋がる「ユビキタスネットワーク社会」が構想されてきた．接続機器として代表的なものとして，従来はパソコンやスマートフォンが挙げられるが，センサーデバイスの普及に伴い，車や家電といった物理機器，建物もネットワークに接続されるようになった．このように様々なデバイスがネットワークに接続されるようになると，それらのデバイス間での情報交換やデータの収集，それに基づく自動化が行われ，新たな付加価値を生むようになる．このような仕組みは Internet of Things(IoT) と呼ばれる仕組みであり，急速に発展している [Atzori et al., 2010][Gubbi et al., 2013a].
IoT は医療，スマートホーム，運送など幅広い分野での適用が検討されているが，収集，処理するデータの増加に伴い．深刻なセキュリティ，プライバシーの問題を引き起こすことが指摘されている [Dorri et al., 2017a]．この問題を解決するために Blockchain(BC) を採用することが検討されている．BC は Bitcoin[Nakamoto, 2008] の中核を為す分散台帳技術のことである．BC はセキュアな取引を保証するが，IoT 分野に適用するにあたり以下のような問題点が存在する．

- 取引の合意形成のアルゴリズムに基づく高いリソース要件
- ブロックサイズに基づくスケーラビリティの悪さ
- 合意形成，および二重支払いを防ぐためのアルゴリズムに基づく遅延

これらは IoT 環境が，現在 BC が利用されている分野とは異なる特性を持つ環境であるためである．IoT 環境に用いられるシステムは，膨大なデータ量やネットワークノード数に対応するスケーラビリティ，プライバシー保護のためのセキュリティ，障害耐性，処理のリアルタイム性などが求められる．そこで，上記の問題を解決するために IoT 環境に適用する目的の分散台帳技術の研究が行われている．スマートホームを題材とし，BC をベースとした新しいアーキテクチャを構成した研究 [Dorri et al., 2017a] では，BC を用いた場合と比べ，パケットと処理のオーバーヘッドを大幅に短縮する

ことを実証している．また，他の例には．BC をベースとしない，有向非循環グラフ (DAG) を用いた Tangle と呼ばれるグラフ構造に基づく IOTA という新たな分散台帳技術を開発し，取引の速度，規模のシミュレーションを行う研究 [Popov, 2016][Kusmierz, 2017] が挙げられる．

IoT 環境を想定した分散台帳技術について研究が行われる一方で，それらの技術を評価する明確な指標は存在しない．BC の性能を評価する研究 [Croman et al., 2016] などは存在するものの，IoT 環境を想定した際にこれらの評価軸を用いることが適切であるとは言えない．本研究は，既存の種々の分散台帳技術を IoT 環境に適用する際，どのような評価指標を用いることが適切であるか議論を行い，その指標に基づき複数の分散台帳技術の性能比較を行う．

本研究の貢献は以下の通りと考える．

- IoT 環境における分散台帳技術の評価指標を提案する．

- 提案した評価指標に基づき既存の分散台帳を評価し，IoT 分野に適用することが適切であるとされる技術の特徴を提示する．

本稿の構成は以下の通りである．第 2 節では，IoT の概略および BC などの既存の分散台帳技術の紹介を行う．第 3 節では，評価指標を提示し，その妥当性についての議論を行う．第 4 節では，定量的な指標を提示する．第 5 節では，4 節で提示した指標に関して分析を行い，その結果を示す．第 6 節では得られた結果から，IoT 環境における分散台帳技術の適用可能性や技術特性などについて議論する．

第 2 章 背景

本章では，IoT についての導入と，本研究で題材とする分散台帳技術についての説明を行う．その後，IoT に分散台帳技術を適用する研究の外観と，その重要性について述べる．

2.1 Internet of Things

Internet of Things(IoT) とは，様々な物理機器などにセンサーやソフトウェアを組み込むことで，情報交換やデータの収集を行えるネットワークを構築する仕組みである．[Vermesan et al., 2009] では，アイデンティティ，物理的属性，および仮想パーソナリティ，知的インターフェースを使用し、情報ネットワークにシームレスに統合されている物理的，もしくは仮想的な”モノ”に存在する標準および相互運用可能な通信プロトコルに基づく，自己構成能力を備えた動的なグローバルネットワークインフラストラクチャとして定義されている．

IoT は、物理的な世界と仮想的な世界を橋渡しすることで，スマートな都市，スマートな工場，資源管理、交通機関、健康、福利厚生など、多くのアプリケーション分野に影響を与える．しかし、ソフトウェアアプリケーションの中で IoT を活用することは、ネットワーキングからアプリケーション層まで，特に超大規模，極端な異質性，IoT の動的性などの大きな課題を抱えていることが指摘されている [Bouloukakis et al., 2016]．また，世界中において配備されているセンサーの数は急速に増加しており，加速度的に増大していくとされる．相互に接続されたデバイスの数は 2013 年の時点で 90 億個であり，2020 年には 240 億個に到達するとみられている [Gubbi et al., 2013b]．これらのセンサーは膨大な量のデータを生成しつづけるが，セキュリティ面に対して十分な対策はなされていない．現在の IoT において，セキュリティとプライバシーを提供する方法を提案するにあたり，3 つ課題が存在すると Dorri ら [Dorri et al., 2017a] は指摘している．1 つ目はリソースの面である．IoT のネットワークにおいてデータを収集するセンサーなどの末端のデバイスは，高度で複雑なセキュリティメソッドを備えていないという問題

がある．2 つ目は，プライバシーの問題である．IoT において収集されるデータは個人が特定されるようなデータが含まれることが想定される．種々のデータを収集しながらユーザーのプライバシーを保護しなければならない．3 つ目は中央化である．クライアントサーバモデルをはじめとする中央集権的なモデルは IoT には不適切であることが多いと指摘されている．中央集権型のモデルを適用した際，単一障害点，多対一トラフィック，スケーラビリティなどの問題を抱えるとされる．

2.1.1 IoT 環境において求められる要件

IoT 環境のシステムに求められる要件は次のようなものが挙げられる．

- 膨大なデータとノードの増加に対応するスケーラビリティ
- プライバシー保護の観点でのセキュリティ
- サイバー攻撃に対し強いこと
- リアルタイムな処理

2.1 節で述べたように，センサーによって収集されるデータ量は膨大なものになる．また，ネットワークに接続されるセンサーの数や種類が急速に増加していくことが考えられるため，拡張性に富む設計であり，大量のデータを問題なく処理することができるスケーラビリティを有することが求められる．
IoT において収集するデータは，工場内のデータや住居内のデータ，個人の身体情報などの機密性が高い情報も含まれる．情報を収集しつつ，これらのプライバシーを保護することのできるシステムが必要である．
また IoT は，都市インフラ，交通などシステムの安定性が求められる分野に利用されることが検討されている．サイバー攻撃や，サーバートラブルに対し強いシステムであることが求められる．
一方，既存のクライアント・サーバモデルでは，中央のサーバの処理速度が全体のボトルネックになることや，中央のサーバがトラブル，攻撃などの被害にあった際にシステムが停止してしまうことといった問題点が指摘されている．

これらを解決するために，中央集権型のシステムではなく，分散データベースを用いることが提案されてきた．分散台帳技術は，既存の IoT 環境のシステムに比べ，セキュリティの面や，攻撃やトラブルに対する耐性に優れる．一方で，現在実用されているブロックチェーンなどの分散型アーキテクチャは処理に時間がかかること，トランザクションの処理速度が遅いこと，トランザクションあたりの手数料が大きいことなどの理由からこれらを直接 IoT 環境に適用することは困難だとされている．

2.2 分散台帳技術

分散台帳技術 (Distributed Ledger Technology) はデータを中央で管理するのではなく，分散させ管理するデータベースであり，複数の端末が同じ情報を保持することで改竄や攻撃への耐性を高め，管理者が存在せずに信頼性を確保するものである．

分散台帳技術のうち代表的であるブロックチェーンは，取引の記録を分散させ，ネットワークの参加者に相互に確認させることで，取引の整合性を高めている．

分散台帳技術のメリットは，不正取引や改ざんが非常に困難である点，またシステムダウンに強い点である．従来の中央集権型のシステムは，管理者が存在し，管理者側に悪意のある人間が存在した場合に不正な取引や改ざんが行われる可能性があるために，管理者の信頼性というものが必要であるが，管理者が信頼できるかどうかユーザーが判断することは困難である．一方で分散台帳技術においては，管理者が存在しないため，管理方式に対してユーザーが安心しやすい．また，中央集権型のシステムの場合，管理者側のサーバーやコンピュータにトラブルが生じた際システムを稼動できない状態に陥る危険性があるが，分散型のシステムの場合は，一部のノードにトラブルが生じてもシステムを稼働し続けることができる．

2008 年，仮想通貨システムの中核をなす技術として公開された分散台帳技術であるが，影響を及ぼしたのは経済の領域に留まらず，エネルギー分野，物流分野，音楽分野，ヘルスケア分野などにおいて革新的なアイデアやサービスの創出に分散台帳技術の可能性が期待されている [**?**].

本研究において分析対象とするいくつかの分散台帳技術を紹介する．

2.2.1 Blockchain

Blockchain(BC) は Bitcoin[Nakamoto, 2008] の中核を為す分散台帳技術である．ブロックと呼ばれるトランザクション (TX) の集合を一定時間おきに生成し，それを鎖のように連結させていくことでデータを保管するデータベースである．右に行くほど新しいブロックであり，鎖状に連結される．トランザクションと呼ばれるユーザ間での取引は 10 分おきにブロックという単位にまとめられ，台帳の最新のブロックの直後に連結される．台帳にブロックを追加するためには Proof of Work(PoW) と呼ばれる作業が必要となる．この作業は多くの計算資源を要求するため，データを保証する根拠となる．PoW を行い，ブロックを追加することができたユーザには報酬が与えられ，この一連の作業はマイニングと呼ばれる．BC の各ブロックは，直前のブロックの内容のハッシュ値と，取引データを含んでいる．そのため，台帳の一部を改竄するには，その後のブロックの内容を全て書き換える必要があり，BC の改竄は実質的に不可能とされている．
Bitcoin のシステムを例として，BC において利用される用語について説明する．

トランザクション トランザクションとは，ブロックチェーンにおける取引記録のことである．トランザクションには，タイムスタンプと，インプット，アウトプットが含まれており，Bitcoin のシステムにおいては，あるユーザーが別のユーザーへ送金したコインの額と，その時間が記録されている．BC 内のトランザクションのアウトプットには，未使用状態の UTXO(Unspent Transaction Output) と呼ばれるアウトプットと，使用済みのアウトプットの 2 種類の状態がある．

ブロック ブロックは，一定時間のうちに生成されたトランザクションの集合からなるデータである．1 つのブロックにはその親のブロックのハッシュ値が含まれており，ブロックを鎖状に繋げてデータを保存していく形式から Blockchain と名付けられている．最初のブロックからあるブロックまでの距離を表現するのに「高さ」，新しく追加された最後のブロックを「トップ」または「先端」と表現する．ブロックチェーンの

ブロックは、自身のヘッダの「previous block hash」のフィールドを通して 1 つ前のブロックを参照しており，参照されているブロックを親ブロックと呼ぶ．親ブロックのハッシュ値を自身のヘッダに持っており，このハッシュ値の連なりを辿っていくと、最終的には genesis ブロックと呼ばれる最初に生成されたブロックに達する，ブロックの構造は以下表2.1のようになっている．

サイズ	フィールド名	内容
4 バイト	ブロックサイズ	次のフィールドからブロックの最後までのデータサイズ
80 バイト	ブロックヘッダ	ブロックヘッダ
1〜9 バイト	トランザクションカウンタ	ブロックのトランザクション数
可変サイズ	トランザクション	ブロックに記録されるトランザクションのリスト

表 2.1: ブロック構造

マイニング マイニングとは，BC において新たなブロックを生成するための行程である．BC のネットワークに参加しているノードはこの作業を行い，マイニングに最初に成功したものがブロックチェーンにブロックを追加することができ，トランザクションを発行する際の手数料を報酬として得ることができる．Bitcoin においてマイニングは 10 分間に 1 度行われる．マイニングは，ブロックヘッダの中のナンス値と呼ばれる値について，ブロックのハッシュ値がある値よりも小さくなるような適切なナンス値を見つけるもので，適切なナンス値の条件を変動させることで，マイニングにかかる時間を一定に保っている．この適切なナンス値を見つける計算は Proof of Work(PoW) と呼ばれる．チェーン中のブロック内のトランザクションの一部を改ざんしようとした際，そのブロックのナンス値を再計算しなくてはならないばかりではなく，そのブロック以降の全てのブロックに対して同様の計算をしなくてはならないため，ブロックの改ざんは不可能に近いとされる．

BC を用いたシステムの挙動を Bitcoin のシステムを例にして順を追って説明する．

1. トランザクションの発行... ビットコインを 1 つ以上のソースアカウントから 1 つ以上の宛先アカウントに転送することでトランザクション

が作成される．送金者は，データ構造としてのトランザクションを作成し，インターネット経由で，Bitcoin ネットワークに接続されている全てのノードに送信する．送金者は，マイニングを行う際に自分のトランザクションをブロックに取り込んでもらうために，手数料を支払う必要がある．この場合，インプットの総額とアウトプットの総額の差額が手数料となる．

2. トランザクションの伝搬... 発行されたトランザクションは，ネットワークのノードからノードへと回送されることでネットワークの隅々までデータとして届けられる．トランザクションは，そのトランザクションを受け取ったノードで内容の生合成の検証をされ，内容が不整合であれば，他のノードへの回送は行われない．

3. 検証... ブロックの作成を行う主体のことをマイナーと呼ぶ．マイナーは自身が受け取ったトランザクションをメモリープールと呼ばれる領域に格納し，マイニングに成功すると，メモリープールの中のトランザクションをブロックに格納していく．Bitcoin において，ブロックサイズの上限は 1MB であり，この上限までしかトランザクションは格納することができない．

4. ブロックの追加... 計算の競争に勝利したマイナーは新しいブロックを作成することができ，この新しいブロックはネットワークにブロードキャストされる．各ノードは，送信されてきたブロックの内容をそれぞれ検証し，問題がない場合自身のノード内の BC に接続する．新しいブロックが Bitcoin ネットワークの全てのノードにおいて BC に接続されると，ブロック内の全トランザクションも非可逆な記録として定着する．ブロードキャストされたトランザクションの伝播には大きなレイテンシが存在する．また，世界中にマイナーは点在しているため，ブロックがほぼ同時に生成されたり，ブロック内のトランザクションもマイナーごとに異なる可能性がある．あるトランザクションがブロックに取り込まれてから最新のブロックが接続されるまでのブロックの数である確認数が 6，つまりあるトランザクションが格納されているブロックに 6 つ新しいブロックが接続されるとトランザクションは非可逆であるとみなされる．

2.2.2 Ethereum

Ethereum[1]は次世代のスマートコントラクトと非中央集権型アプリケーションのプラットフォームと表現されている，独自 BC 上でアプリケーションを配置し，実行できるプラットフォームである．Ethereum はスマートコントラクトを記述する Solidity という言語に代表されるチューリング完全なプログラミング言語を持ち，ユーザは Ethereum 上の BC に任意の分散型アプリケーションやスマートコントラクトを記述し実行することが可能である．

スマートコントラクト スマートコントラクトは，1994 年に Nick Szabo により「契約を実行するコンピュータ化されたトランザクションプロトコル」として定義された [Szabo, 1994]．また，担保，譲渡といった契約条項をコードに翻訳し，自己執行可能なハードウェア，あるいはソフトウェアにそれらを埋め込むことを提案した [Szabo, 1997]．これにより，仲介者の必要性を最小限に抑え，信頼性のある取引が可能となる．ブロックチェーン上のスマートコントラクトでは，契約をプログラムで定義し，条件に合致した際には仮想通貨などのデジタル情報で取引を行い，契約が正当に履行されたかの確認も自動化して実行する．決済や権利の取引の自動化にスマートコントラクトを活用できれば，これらの取引のプロセスを P2P で実行でき，仲介者が不要となるため，自律的で透明性の高い取引が行うことが可能となると期待されている．ブロックチェーンを利用したスマートコントラクトは，過去の実行履歴がブロックチェーン上に記載されるため，取引の改ざんを防ぐことが可能である．金融・証券や，生活・産業など契約と密接に関連する業務への適用が検討されている．

Ethereum で使う通貨は「Ether(イーサ)」と呼ばれる．Ethereum での送金，コントラクトの実行には「Gas(ガス)」と呼ばれる手数料が必要となる．また，Ethereum はコンセンサスアルゴリズムとして Bitcoin と同じ PoW を用いている．Bitcoin では最も長いブロックチェーンを有効なチェーンとみなすのにたいし，Ethereum では最も多く計算量が投下されているブロックチェーンを有効なチェーンとする GHOST(Greedy Heaviest Observed Subtree) プ

[1] https://www.ethereum.org/

ロトコルが採用されている．これは，Bitcoin のブロック生成時間が 10 分間なのに対し，Ethereum のブロック生成時間は 15 秒と短く，チェーンのフォークが頻繁に起こるためである．

Ethereum のコンセンサスアルゴリズムは，Proof of Stakes(PoS) にアップデートされることが計画されている．PoS においてブロック生成に参加している人々はバリデーターと呼ばれる．バリデーターがステイクしている通貨の量に応じて，ブロックごとに抽選でブロックを生成する権利が与えられ，ブロックを生成したバリデーターに生成報酬が与えられる．

ブロックガスリミット Ethereum の特徴として，Bitcoin のようにブロックサイズの上限が明確に定められていないことがある．マイナーはブロックガスリミットと呼ばれる値を，投票によってマイブロックごとに変動させることができ，この概念が Bitcoin におけるブロックサイズに似た役割を果たしている．Ethereum ではトランザクションの処理を行うために Gas を支払うが，1 ブロックに取り込むことのできる Gas の上限がブロックガスリミットである．

2.2.3 IOTA

IOTA[2]は M2M マイクロペイメントのために開発された仮想通貨の基盤となる分散台帳技術である．IOTA は BC を IoT に適用する際に課題とされる，スケーラビリティとトランザクションフィーを解決するために開発された．第三者の関与なしに，マシンが相互にサービスを提供し，電気，ストレージ，データといったリソースをトレードすることを想定し，これらのために軽量で，スケーラビリティに富んだ分散台帳が必要であると述べている．ユースケースとして，オランダにおいて IOTA を用いて課金と支払いができる充電スタンドが設置されている．これは自律的に電気自動車の充電を行うための通信や支払いを担い，メーターの値を台帳に保存することで信頼性が高い管理を保証するものである．

IOTA は BC とは異なり，Tangle と呼ばれる分散台帳アーキテクチャを用いている．Tangle は有向非循環グラフ (Directed Acyclic Graph) に基づいており，BC のようなブロックの概念は存在しない．IOTA において，ネットワークの参加者は BC と異なり，台帳に任意のタイミングでトランザク

[2]https://www.iota.org/

ションを追加することができる．この際，Tangle に追加されているトランザクション (TX) のうち，2 つを選択して PoW を行い，それらのトランザクションの正当性を証明することで，自分のトランザクションをネットワークに追加することが可能になる，図2.1の場合，8 のトランザクションをグラフに追加するために 5，6 の 2 つのトランザクションを検証したということを表している．また，単位時間あたりのトランザクションが増加しても未承認のトランザクション（グラフの左端のトランザクション）の数は発散せず一定であり, 安定性があるとされている [Kusmierz, 2017].

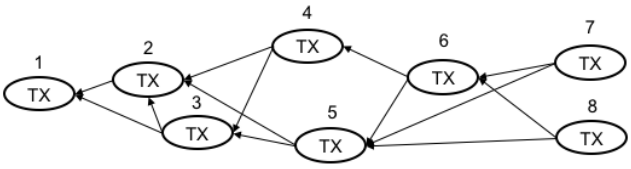

図 2.1: IOTA 概略

Tangle について詳細な説明を与える．Tangle は IOTA に利用されている有向非循環グラフであり，発行されたトランザクションは Tangle 上で集合を形成する．新しいトランザクションが到着すると，前の 2 つのトランザクションを承認する必要がある．これらの承認は有向エッジによって表される．トランザクション A とトランザクション B の間に有向エッジがなく，A から B までに少なくとも長さ 2 の有向パスがある場合，A は B を間接的に承認する，と定義されている．ノードはトランザクションを発行するために，以下のことを行う必要がある．

- アルゴリズムに従って，承認する 2 つの他のトランザクションを選択する．
- ノードは 2 つのトランザクションが矛盾していないかどうかを確認し，矛盾しているトランザクションは承認しない．
- 承認されたトランザクションのいくつかのデータと鎖状に繋がったナンス値のハッシュがある条件を満たすような，ナンス値を見つける作業を行う．これは Bitcoin の BC で行われる Proof of Work と同様なものである．

Tangle において，トランザクションが承認されるには，「全ての未承認のトランザクションから直接的あるいは間接的に承認される」ことが必要となるが，これは容易に確認することができる．

IOTA は，BC や Ethereum と異なり，ブロックを生成することがなく，トランザクションの発行者自身が PoW を行い報酬として台帳に自身のトランザクションを追加する権利を得る．そのため，トランザクションを発行するために BC や Ethereum では必要となった手数料が必要ではない．このため，1 回あたりの取引の金額が少額になることが想定される IoT 環境において適しているとしている．また，BC や Ethereum のように一定の間隔で承認が行われるのではなく，並列して行われるため，ネットワークに大量のトランザクションが追加されるほどトランザクションは承認されやすくなり，スケーラビリティにも富むとされている．

IOTA の承認周りの話をもう少し詳しく行う．

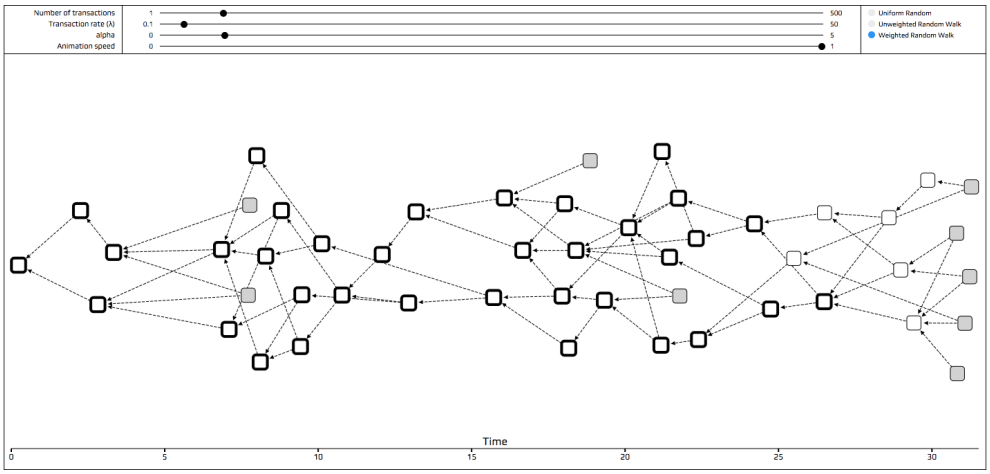

図 2.2: Tangle 概略

2.3 IoT 環境における分散台帳技術

IoT 環境では，現在より膨大な量のデータのやりとりが行われるようになる．Croman らは，現在の BC の性能評価を行い，理論的に最大 1 秒あたり 7 トランザクションの処理が可能であるものの，実際の BC における測定では 1 秒あたり 3.3 トランザクション程度の性能であると述べている [Croman et al., 2016]．一方で，現在の VISA の支払いストリームは毎秒平均

で 2000 トランザクションの処理が行われており，BC のスケーラビリティとの大きな乖離が指摘されている．また，BC は仕様上，10 分に 1 度まとめてトランザクションの合意を行うため，リアルタイムの処理を行うことはできない．IoT，金融などのシステムではリアルタイムの処理が要求されるため，この点においても現状の BC を転用することは困難であると指摘されている．

また，[Decker and Wattenhofer, 2013] では，台帳を共有するネットワークのメッセージ交換がオーバーヘッドになることを指摘している．各ノードが新規ブロックを受け取るまでの遅延は平均で 12.6 秒であり，ブロックサイズが大きくなればなるほど，遅延時間も比例して大きくなるとしている．

Dorri らは，IoT 環境としてスマートホームの環境を想定し，既存の BC をベースにした新たな軽量なアーキテクチャを提案し，セキュリティ，プライバシーに対し BC ベースのアーキテクチャの有効性について分析を行った [Dorri et al., 2017b]．また，パケットオーバーヘッド，時間オーバーヘッド，エネルギー消費の観点から提案アーキテクチャを評価，議論した．

実際に IoT 環境に分散台帳技術を適用する際の IoT システムの概要図を図2.3に示す．

IoT 環境の分散台帳技術において求められる要件は，既存の分散台帳技術に求められる要件と重要視される点が必ずしも等しくないと考えられる．前節で提示した評価指標において，本研究は IoT 環境における分散台帳技術の評価指標を作成することを考慮すると，性能効率性，スケーラビリティの 2 点がより重要視されると考えられる．これは，IoT 環境においては Bitcoin などのシステムに比べ即時的な処理が要求されること，ノード数，データ数が非常に膨大になることが想定されるためである．

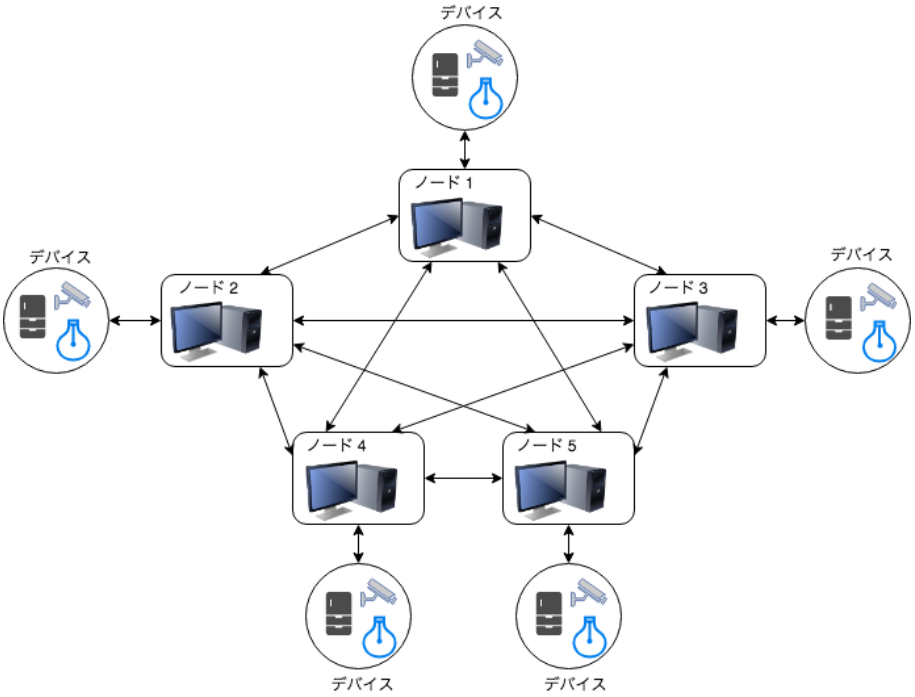

図 2.3: IoT 環境概要

第 3 章 分散台帳技術の性能指標

本研究は，IoT 環境に分散台帳技術を適用する際，それらの性能を比較するための適切な評価指標を提示し，既存のアーキテクチャに対し比較を行い，どのようなアーキテクチャが ioT 環境に適しているか考察する．既存研究において，BC の性能評価，IoT への適用可能性や具体的なシナリオにおける評価などは議論されてきたが，IoT 環境に分散台帳技術を適用する際どのような性能評価が必要であるかという点の議論は未だ不十分である．分散台帳技術の評価を行うにあたり，ネットワーク，サービスコンピューティングなどの観点も含め，以下表3.1のような評価項目が考えられる．

説明

3.1 一般的な指標

一般的な性質の分析は主に，それらの構造的特徴を捉えるものである．シミュレーションによる分析に対し，種々のシステムの差異がアーキテクチャ，アルゴリズムなどのどの要素によって生じ得たものか検討する際に要求されるものである．図3.1の関連技術の項目が各分散台帳技術の性能，特性差を生むものであるため，一般的な性質の分析指標としてこれらの項目を取り上げる．

- ネットワーク環境
- ブロックサイズ
- トランザクションサイズ
- コンセンサス方式
- ブロック生成時間

これら 5 点が性能面に対して具体的にどのような影響を与えるか説明する．ネットワーク環境は，具体的にはユーザが自由にネットワークに参加することができるかということを指す．IoT 環境においては，日々ネットワークへの参加ノード数が増大していくことが考えられるため，管理者によ

大項目	概要	小項目	関連技術
性能効率性	システムの応答時間および処理時間並びにスループット速度が要求事項を満足する度合い	処理性能	ブロックサイズ トランザクションサイズ コンセンサス方式 ブロック生成時間
		ネットワーク性能	ネットワーク環境 ノード分散
		ブロック確定性能	コンセンサス方式 ネットワーク環境
		参照性能	ノード分散 ネットワーク環境 ブロック構造
スケーラビリティ	処理速度を向上させられる度合い，保持するデータ量の増大に対する拡張性の度合い，対応可能ノード数など	スループット向上性	ブロックサイズ トランザクションサイズ コンセンサス方式 ブロック生成時間
		ネットワーク性能向上性	ノード分散 ネットワーク環境 P2P プロトコル
		容量拡張性	ブロックサイズ トランザクションサイズ コンセンサス方式 ブロック生成時間
		ノード数拡張性	データ容量 コンセンサス方式
信頼性	運用操作可能及びアクセス可能な度合い	可用性	単一障害点の有無 コンセンサス方式
セキュリティ	アクセスすることを認められたデータのみにアクセスできる度合い	機密性	アクセス管理 データ秘匿化
	行為が引き起こされたことを証明することができる度合い	否認防止性	コンセンサス方式

表 3.1: 評価軸

ってネットワークへの参加者が制限されるシステムは適さないと考えられる．しかし，管理者がネットワークに参加するユーザを事前に制限することで，ある程度のセキュリティが確保できるため，トランザクションの検証のための PoW を簡素化することができるメリットはある．PoW の簡素化は，トランザクションあたりのコストの減少と処理速度の向上が見込まれるため，スマートファクトリーやスマートホームといった限られたネットワーク内での利用には有用である可能性がある．ブロックサイズは，台帳のブロック 1 つあたりの容量を指す．ブロックサイズが大きくなれば，1 回のブロック生成の際に格納することのできるトランザクションの数が増大するため，ブロック生成時間が一定であるとするならば単位時間あたりのトランザクションの処理数が増大する．一方で，ブロックサイズが増加す

ればするほど，ブロックがネットワークを伝播する際の遅延が大きくなるため，ブロックのフォークが生じやすくなる．また，IoT 環境において，末端のデバイスの通信速度が常に確保できるとは限らないため，この遅延がより大きくなることも想定される．トランザクションサイズは 1 取引あたりのデータ量を表す．トランザクションサイズが小さくなれば，ブロックサイズを一定とした際 1 ブロックあたりに格納できるトランザクションの数が増大するため，単位時間あたりのトランザクションの処理数が増大する．一方で，トランザクションサイズを小さくすることは，ハッシュ等の簡略化を引き起こし，セキュリティ面での低下をもたらすことが考えられる．コンセンサス方式は，新たなトランザクションを台帳に追加する際にどのような合意形成の方法を取るかを指す．代表的なものとしては Bitcoin などに用いられている PoW，Ethereum に導入される PoS などがある．コンセンサス方式によって，合意形成に必要な時間，必要となるリソース，セキュリティなどが変化すると考えられる．ブロック生成時間は，あるブロックが作成されてから次に新たなブロックが作成されるまでの時間である．ブロック生成時間が短くなれば，1 ブロックあたりに格納されるトランザクションの数を一定とした際，単位時間あたりのトランザクションの処理数が増大する．一方で，ブロック生成時間を短くするということはコンセンサスに要する時間を短くすることと同義であり，検証の時間が短くなればなるほどセキュリティの低下を引き起こす．また，ブロックの伝搬にかかる遅延に対して生成間隔が短くなると,チェーンのフォークが起こりやすくなる．

2 章で説明した 3 つのシステムを，上記の指標に指標に基づいて比較した．以下図3.2に示す．

	BC	Ethereum	IOTA
ネットワーク環境	誰でも参加可能	誰でも参加可能	誰でも参加可能
ブロックサイズ	1MB	不定	-
トランザクションサイズ	500byte	可変	1600byte
コンセンサス方式	PoW	PoW/PoS	PoW
ブロック生成時間	10m	15s	不定

表 3.2: 一般的な性質の分析

ネットワーク環境においては，本研究において対象とした 3 種ともに誰

でもネットワークに参加可能なパブリックなネットワークである．ブロックサイズについては，BC が 1MB と固定値であるのに対し，Ethereum は不定である．Ethereum においてブロックサイズに相当する概念はブロックガスリミットと呼ばれ，4,5 ヶ月の単位で変動する．IOTA に関してはブロックを生成しない．トランザクションサイズは，BC が平均して 500 バイト [Croman et al., 2016]，IOTA が 1600 バイトとされている．Ethereum においてトランザクションサイズに相当する概念はガスリミットと呼ばれ，トランザクションを実行する手数料としてユーザーが指定できる．コンセンサス方式は，3 種とも PoW であり，計算量を要する問題を最初に解いた者がブロック（トランザクション）を追加することができる．Ethereum に関しては PoS を今後採用することが検討されている．ブロック生成時間は BC が 10 分であるのに対し，Ethererum が 15 秒である．Ethereum はブロックの生成間隔が大幅に短い分，BC に比べチェーンのフォークが起こりやすい．IOTA は PoW が完了した段階でトランザクションを Tangle に追加するため，ブロック生成時間は定まらない．トランザクションが承認されるまでの時間についても，BC と Ethereum はブロック生成時間に一致するが，IOTA はトランザクションレートなどにより変動する．

3.2 シミュレーションにより測定する指標

本節では，分散台帳技術を評価するための指標のうち，シミュレーションにより測定する指標について定義し，詳細に説明する．指標としては，大別して速度，スケーラビリティの 2 つを定める．

3.2.1 速度

速度 (遅延) は，主に分散台帳技術の処理性能に大きく関わる点である．現状利用されている分散台帳技術を用いたシステムと比較し，IoT 環境を想定したシステムでは，発生するトランザクションの数が膨大になると考えられる．そのため，IoT 環境ではより高い処理性能が求められる．また，IoT 環境ではリアルタイムな処理が求められることがある．これらのことから，遅延やトランザクションの処理にかかる時間は短ければ短いほど良いシステムであると言える．

本研究においては速度 (遅延) を示す指標として，以下の 2 点について評価する．

- ノード間遅延
  ノード間遅延は，あるノードにおいてトランザクションが発行されてからそのトランザクションがネットワーク内の全ノードに伝搬し，全ノードから確認できるようになるまでにかかる時間と定義する．これは主にネットワークの通信速度に依存する．ノード間遅延が長くなると，ノード間の不整合が生じやすくなり，システムの防御性が弱まる結果を招く．

- 承認時間
  承認時間は，発行されたあるトランザクションが承認されるまでの時間と定義する．IoT 環境においては処理のリアルタイム性が求められる場合もあるため，承認時間が長くなると，システムとして適切ではない．

3.2.2 スケーラビリティ

スケーラビリティは処理するデータ量の増大及びネットワークの参加者の増大に対する拡張性の度合いである．生成されるトランザクションのスピードがどの程度であればシステムが正常に動作することができるか，データ量やネットワークの参加者が増大した際にネットワーク遅延がどの程度生じるかという点を指標とすることで，データ及びノードが膨大となる IoT 環境を想定した評価指標になると考える．

- キャパシティ
  キャパシティは，単位時間あたりに処理できるトランザクション数と定義する．現状，分散台帳技術を用いたシステムは，トランザクション数が大規模になる状況に適していない．あるシステムのキャパシティが大きければ大きいほど，トランザクション数が膨大になることが想定される IoT 環境において利用するシステムとして適切であると考えられる．

- ノード規模耐性

ノード規模耐性は，ノードの規模が増大することに対してシステムの承認時間やキャパシティがどのように変動するかというものである．

3.2.3 処理達成率

処理達成率は，発行された全トランザクションのうち，ある時点までにどの程度の割合のトランザクションが承認されているかということを示すものである．ノード規模，秒間に発行されるトランザクションの数が増大しても，処理達成率に大きな変化が起きないシステムは，スケーラビリティがあると考えられる．

詳細化がそれぞれ必要になる

第 4 章 性能分析のためのシミュレーション設計

本研究では，3 章で提案した評価指標に基づき，既存の分散台帳技術の性能比較を行う．そのために提案指標を測定するシミュレーションを行う．本章では，性能比較のために設計したシミュレーションについての詳細な説明を与える．

4.1 シミュレータ設計

本研究は Ethereum について性能を分析する．そのためにシミュレータを設計する．
ネットワークは，図4.1のように複数のノードが他のノードにそれぞれ接続され，ネットワークを構成しているモデルを構成する．ネットワークのノード数は $|N|$ で表され，図4.1は $N = 5$ の場合である．シミュレーションにおいて，ノード 1 からその他のノードに同数ずつトランザクション $|T|$ を発行する．これらのノードは，2 章における図**??**の各ノードに対応し，ノードとデバイスのやりとりがトランザクションに対応する．例として図4.4を示す．図4.4は $N = 5$，$T = 5$ の場合であり，結果としてネットワーク中に 20 トランザクションが発行されたことになる．

本研究では，$N = \{2, 5, 11, 21\}$，$T = \{20, 60, 100, 200\}$ についてシミュレーションを行う．
Amazon EC2 上で Docker を用いて仮想的に台帳を共有する複数のノードを起動する．使用したマシンイメージは Amazon Linux 2 AMI (HVM), SSD Volume Type であり，インスタンスタイプは t2.xlarge（4CPU，メモリ 16GB）である．下記の手順で環境の構築を行う．

1. nodeCountAndFirstId.sh の編集
   ファイル内に変数 nodeCount が定義してある．この値を変更すると，起動させるノードの数が変更できる．

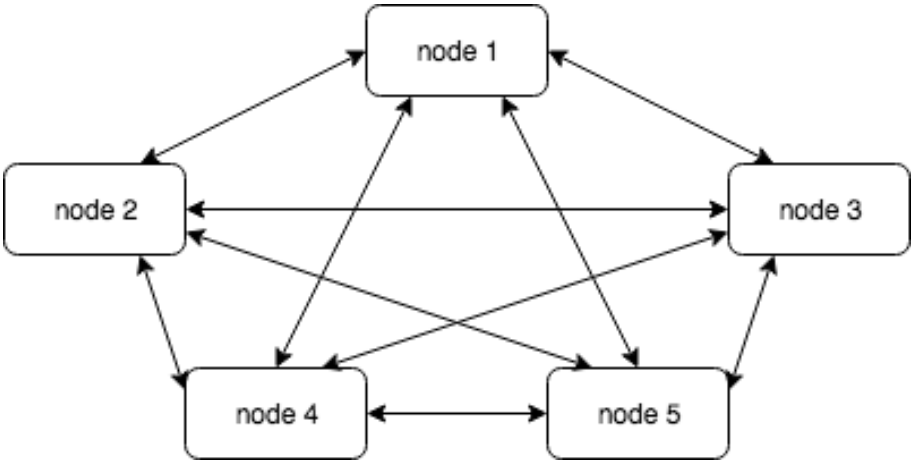

図 4.1: シミュレーションモデル概要

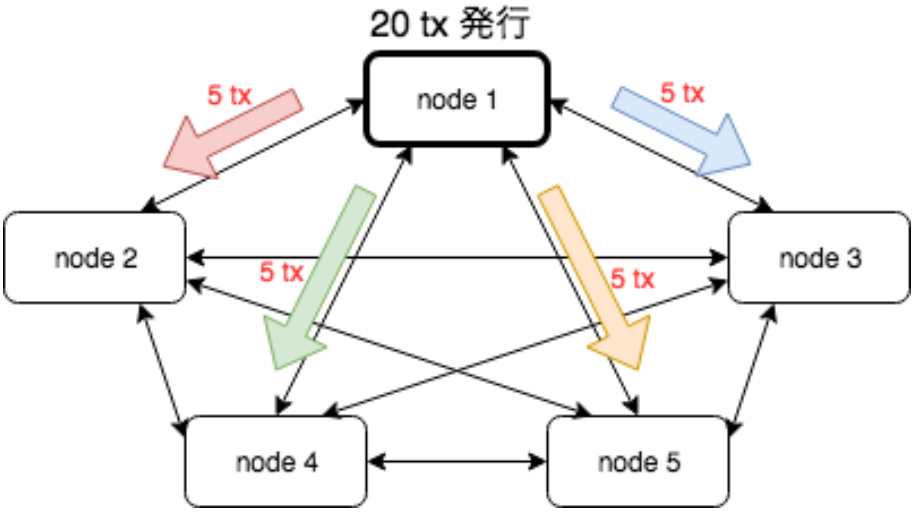

図 4.2: トランザクション発行

2. 01_setupNodes.sh の実行

   このファイルは Docker を呼び出して仮想的なネットワークを作成し，そのネットワーク内に 1. で指定された数のノードを起動する．各ノードに 1 つずつアカウントが作成される．起動したノードは他の各ノードと接続される．最初に起動したノードでマイニングを行い，トランザクション発行のための eth を獲得する．貯まった eth は loginEthereum1.sh で確認することができる．

4.2 分析指標

前節で構築したシミュレータによって分析を行う指標についてより詳細に定義する．

4.2.1 トランザクションの伝播時間

トランザクションがノード 1 から発行されたのち，ネットワーク中を伝播し，ネットワーク内の全ノードに到達するまでの時間を測定する．

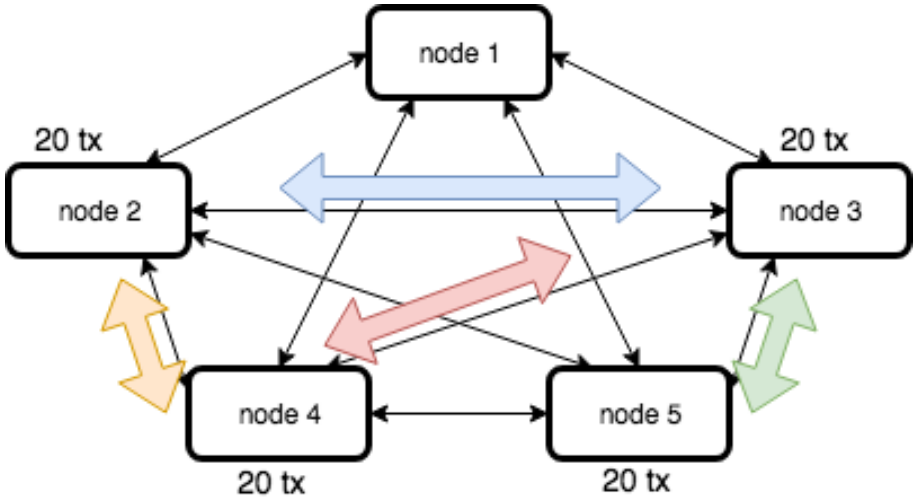

図 4.3: トランザクション伝播

4.2.2 トランザクションの承認時間

トランザクションがあるノードから発行されたのち，そのトランザクションが承認され，ネットワーク内のノードから承認されたことが確認されるまでの時間を測定する．これは

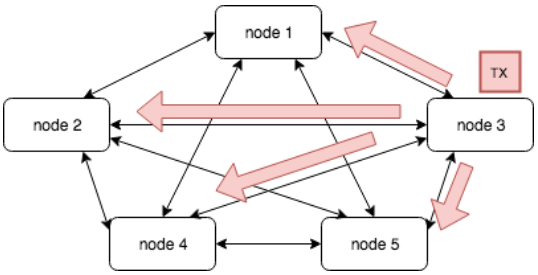

図 4.4: ブロックの作成

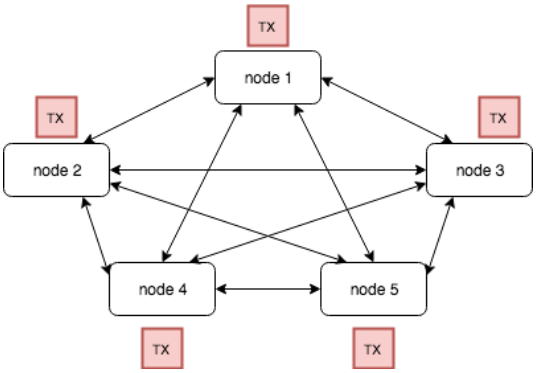

図 4.5: ブロックの伝播

4.2.3 処理達成率

一定の数のトランザクションが一定間隔で生じる環境を想定し，ある時点で発行されたトランザクションのうちどれだけが処理されたとみなせるかということを考える．トランザクションが承認されるまでの処理を複数回繰り返し行い，時系列のデータとして扱う．

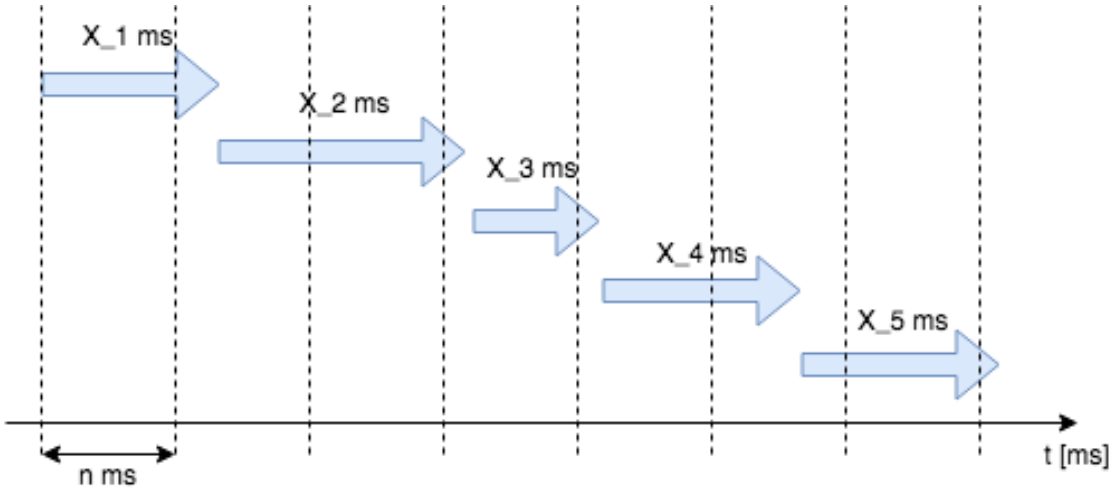

図 4.6: 処理達成率概要

図4.6に基づいて詳細に説明する．図4.6において，$n$ms がトランザクションが発行される間隔である．図の青矢印はその一定個数のトランザクションの承認に要した時間である．1 回目の承認に要した時間が X_1，2 回目の承認に要した時間が X_2 である．$t = n$ において，$t = 0$ に発行されたトランザクションは承認されていない．よって，達成率は $0/1 = 0\%$ とできる．$t = 2n$ において，$t = n$ に発行されたトランザクションは承認されていないが，$t = 0$ に発行されたトランザクションは承認されていると考えられる．よって，達成率は $1/2 = 50\%$ とできる．これを $t = 100n$ まで行い，処理達成率の変動を確認する．また，(発行トランザクション数) - (承認トランザクション数) = (未承認トランザクション数) であり，この未承認トランザクション数も処理達成率と同時に変動を確認する．
本研究ではトランザクションレートとして

- low ... 5tx/s
- middle ... 10tx/s
- high ... 20tx/s

の 3 つを想定し，それぞれ測定を行う．

第 5 章 シミュレーションによる分析

本章では，3 章で提案した評価指標に基づき，既存の分散台帳技術の性能比較を行う．本研究では，3 章で提案した指標のうち，速度およびスケーラビリティを測定するために，4 章に基づきシミュレータを構築し，シミュレーションを行った．

5.1 トランザクションの伝播時間

図5.1，図5.2，図5.3に 2 ノード，5 ノード，11 ノードでのシミュレーションの結果を示す．また，これらの結果の平均値 (AVE) の数値をノード数，トランザクション数についてまとめたものが図5.4である．

	20tx	60tx	100tx	200tx
AVE	731.71	1151.68	1682.49	2598.32
MIN	342	571	815	1364
MAX	4042	1932	2501	4638
tx/ms	36.5855	19.19466667	16.8249	12.9916

図 5.1: 2 ノード

	20tx	60tx	100tx	200tx
AVE	1941.67	3646.23	4424.17	13640.49
MIN	579	1145	1531	2606
MAX	3810	7295	10217	427780
tx/ms	97.0835	60.7705	44.2417	68.20245

図 5.2: 5 ノード

	20tx	60tx	100tx	200tx
AVE	1917.6	5288	8677	17484.8
MIN	1565	2525	3443	5647
MAX	3093	13120	16183	29112
tx/ms	95.88	88.13333333	86.77	87.424

図 5.3: 11 ノード

	2node	5node	11node
20tx	731.71	1941.67	1917.6
60tx	1151.68	3646.23	5288
100tx	1682.49	4424.17	8677
200tx	2598.32	13640.49	17484.8

図 5.4: トランザクションの伝搬時間

図5.4を元に，ノード数ごと，トランザクション数ごとの比較を行った．以下図5.5，図5.6に示す．
図5.5より，トランザクション伝播において，同トランザクション数でもノード数が増加するにつれ，ノード数の増加に比例して伝播時間が長くなることが分かる．線形？
また，図5.6より，同ノード数において，トランザクション数が増加するにつれ伝播時間が長くなることが分かる．
これらは伝播時間 T_s がノード数 N，トランザクション数 M，1 つのトランザクションの伝播時間 T として $T_s = \sum_i^N \sum_j^M T_{ij}$ と表せることを実測的に示している．ここについては検討

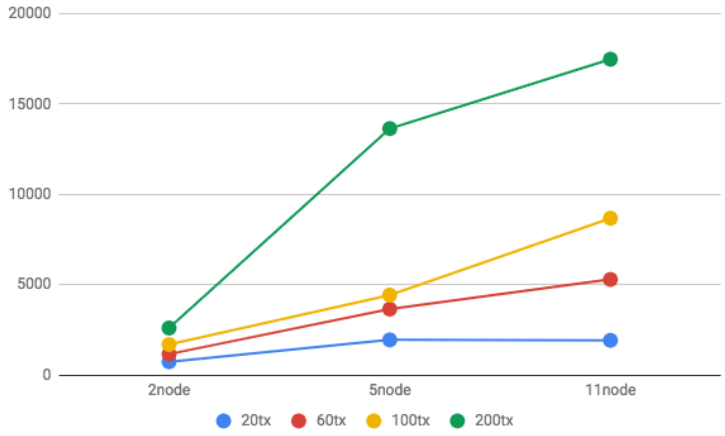

図 5.5: ノード数ごとの比較

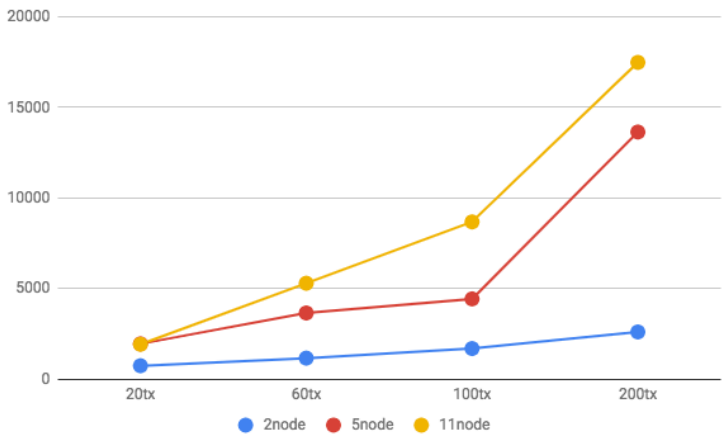

図 5.6: トランザクション数ごとの比較

5.2 トランザクションの承認時間

図5.7，図5.8，図5.9に 2 ノード，5 ノード，11 ノードでのシミュレーションの結果を示す．また，これらの結果の平均値 (AVE) の数値をノード数，トランザクション数についてまとめたものが5.10である．

	20tx	60tx	100tx	200tx
AVE	7579.7	18092.83	17879.7	24284.1
MIN	382	923	2142	1762
MAX	55295	125335	111415	124377
tx/ms	378.985	301.5471667	178.797	121.4205

図 5.7: 2 ノード

	20tx	60tx	100tx	200tx
AVE	1941.67	3646.23	4424.17	13640.49
MIN	579	1145	1531	2606
MAX	3810	7295	10217	427780
tx/ms	97.0835	60.7705	44.2417	68.20245

図 5.8: 5 ノード

図5.10を元に，ノード数ごと，トランザクション数ごとの比較を行った．以下図5.11，図5.12に示す．データ収集
5.11より，$T = 60 T = 100$ について，ノード数に関わらずあまり承認時間

	20tx	60tx	100tx	200tx
AVE	1917.6	5288	8677	17484.8
MIN	1565	2525	3443	5647
MAX	3093	13120	16183	29112
tx/ms	95.88	88.13333333	86.77	87.424

図 5.9: 11 ノード

tx	2node	5node	11node
20tx	7579.7	4057.67	11928.2
60tx	18092.83	19989.12	13436
100tx	17879.7	13953.85	11923.2
200tx	24284.1	19576.05	18648.8

図 5.10: トランザクションの承認時間

の変化が見られなかった．これは，このノード規模において，発行されたトランザクションの伝播時間，生成されたブロックの伝播時間に比べてブロックの生成間隔が長く，$T = 60 T = 100$ についてブロックの生成数に差がなかったため承認時間に大きな差がなかったものと考えられる．一方で，$T = 20, T = 200$ については承認時間は大きく増加している．
5.12より，同トランザクションの条件でノード数による大きな差は見られなかった．これも，前述したように承認時間に大きく影響するブロックの生成数はトランザクションの数によるものであるためである．

5.3 処理達成率

処理達成率と，未承認トランザクション数について測定した結果を以下に示す．$\{N, T\} = \{5, 20\}, \{5, 100\}, \{11, 100\}, \{11, 200\}$ について，トランザクション数ごとに結果を示す．

5.3.1 $T = 20$

$N = 5, T = 20$ の結果について，以下図5.13，図5.14に示す．
トランザクションレートが low のときは，処理達成率が 1 に近く，未承認トランザクション数もほぼ 0 のままであり，発行されるトランザクションが

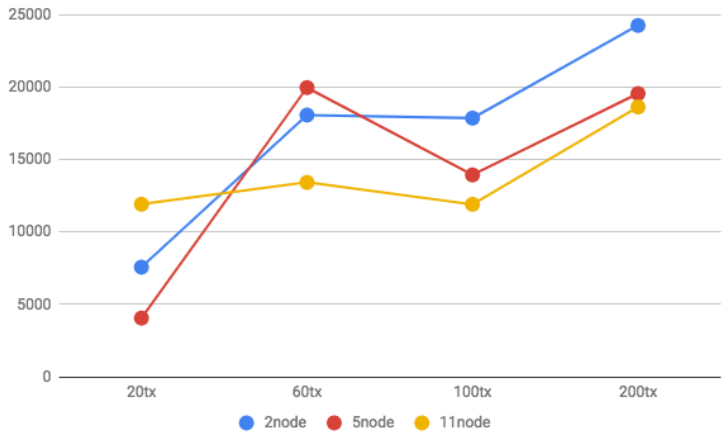

図 5.11: ノード数ごとの比較

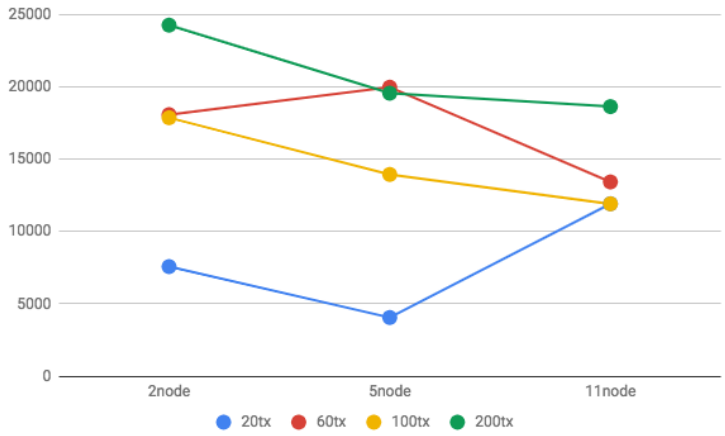

図 5.12: トランザクション数ごとの比較

ほぼ全て処理されていることが分かる．一方で，トランザクションレートが middle のときは処理達成率が 0.5 程度，high のときは 0.25 程度に落ち込み，未承認トランザクション数が線形に増加していくことが分かる．

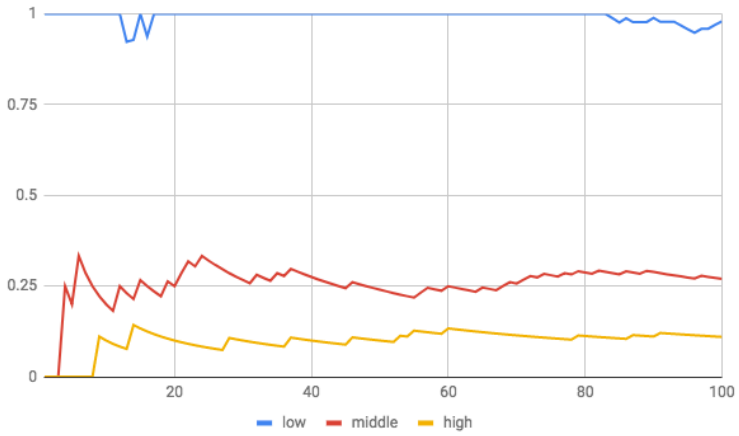

図 5.13: 処理達成率 ($N = 5, T = 20$)

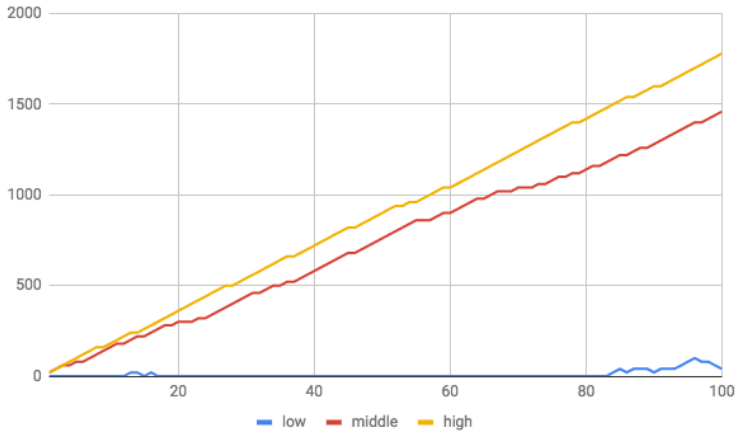

図 5.14: 未承認トランザクション数 ($N = 5, T = 20$)

5.3.2 $T = 100$

$N = 5, T = 100$ の結果を，図5.15，図5.16に，$N = 11, T = 100$ の結果を図5.17，図5.18に示す．

$N = 5, N = 11$ ともに，トランザクションレートが low のときは，処理達成率が 1 に近く，未承認トランザクション数もほぼ 0 のままであり，発行され

るトランザクションがほぼ全て処理されていることが分かる．一方で，トランザクションレートが middle のときは処理達成率が 0.5 程度，high のときは 0.25 程度に落ち込み，未承認トランザクション数が線形に増加していくことが分かる．

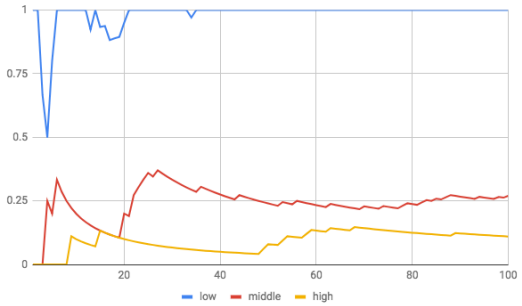

図 5.15: 処理達成率 ($N = 5, T = 100$)

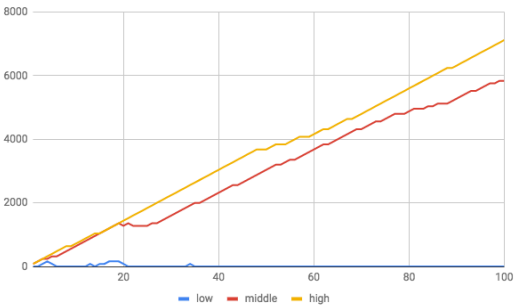

図 5.16: 未承認トランザクション数 ($N = 5, T = 100$)

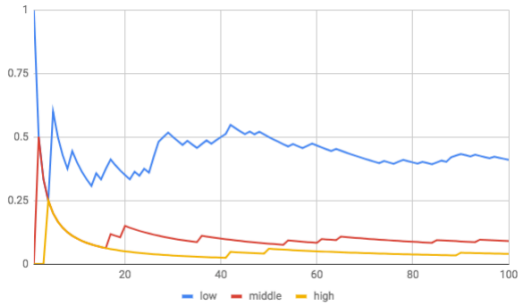

図 5.17: 処理達成率 ($N = 11, T = 100$)

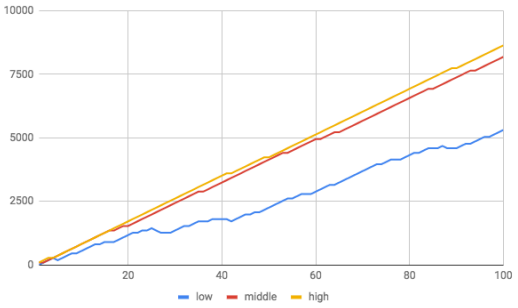

図 5.18: 未承認トランザクション数 ($N = 11, T = 100$)

5.3.3 $T = 200$

$N = 5, T = 20$ の結果について，以下図5.19，図5.20に示す．
トランザクションレートが low のとき，処理達成率は 0.3 程度であり，発行されるトランザクションが十分に処理されていないことが分かる．以降，トランザクションレートが高くなっていくごとに処理達成率は低下していき，トランザクションレートが high のときの処理達成率は 0.1 を下回った．

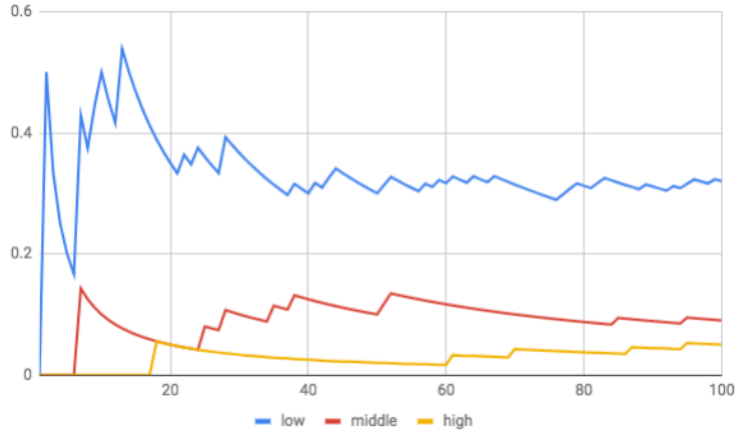

図 5.19: 処理達成率 ($N = 11, T = 200$)

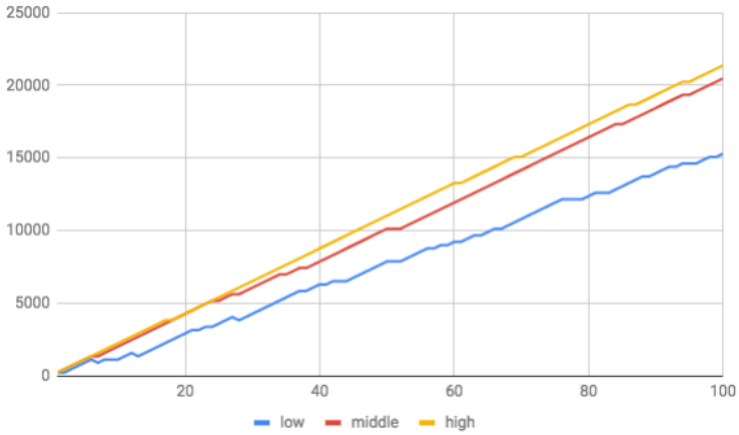

図 5.20: 未承認トランザクション数 ($N = 11, T = 200$)

第 6 章 考察

本章では，第 4 章，第 5 章において行った分散台帳技術の分析に基づき，IoT 環境に分散台帳技術を適用する際に適切である技術特徴について考察を行う．

それほどトランザクションレートの高くない環境においては，IOTA のような形式だといつまでも承認されないトランザクションが存在することになってしまうので，Ethereum のような一定時間ごとにトランザクションが承認されるような形式であることが望ましい．一方で，トランザクションレートが高い環境においては，Ethereum のような形式では未承認トランザクションが増加してしまうため，適さないということが考えられる．IoT 環境においては，計算資源の問題でエッジサーバーが分散台帳のノードの役割を果たすことが考えられるが，トランザクション数が等しい場合，ノードの数の増加が処理能力に影響を与えるため，エッジサーバーあたりのデバイス数を多くし，エッジサーバーの数を減らすことが処理能力の向上のためには有効であると考えられる．

・トランザクションの伝搬速度についてトランザクションの伝搬にかかる時間は，ノード数が等しい場合は発行されるトランザクションの数が多くなればなるほど多くなり，トランザクション数が等しい場合はノード数が多くなればなるほど大きくなる．・トランザクションの承認速度についてトランザクションの承認にかかる時間も同様に，トランザクションの数が多くなればなるほど多くなり，トランザクション数が等しい場合はノード数が多くなればなるほど大きくなる．トランザクションの数が大きくなる影響に比べて，ノードの数が大きくなることによる影響が大きい．ブロックを承認した後それらを全ノードで確認する作業が入るため？· トランザクションの処理完成率について Ethereum については，最初のほうはある程度完成率が変動するものの，時間が経つにつれて割合としてはある値に収束する．完成率が収束するため，未承認のトランザクション数は線型的に大きくなっていくということがわかる．IOTA については，未承認のトランザクション数が一定値に近づくということが考えられるため，トランザクシ

ョンの承認率は 1 に近づいていくことが考えられる (まだ未確認)

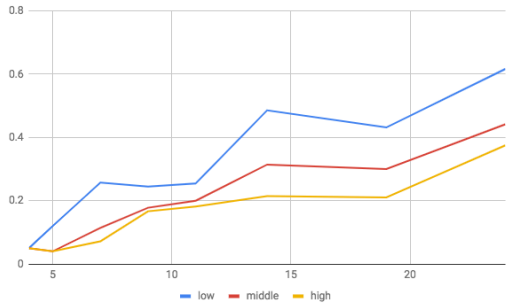

図 6.1: Tangle の処理達成率

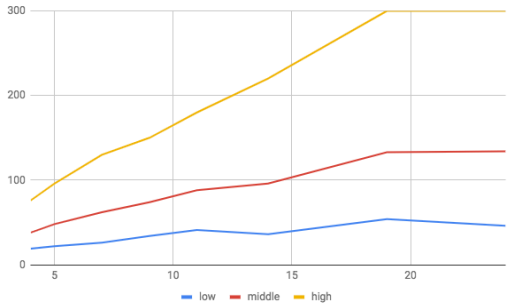

図 6.2: Tangle の未承認トランザクション数

第 7 章 まとめ

本研究では，既存の種々の分散台帳技術を IoT 環境に適用する際，どのような評価指標を用いることが適切であるか議論を行い，その指標に基づき複数の分散台帳技術の性能比較を行った．性能比較のため，IoT 環境を考慮し，分散台帳技術の評価基準のうち性能効率性とスケーラビリティに重点を置いた指標を作成した．また，トランザクションの伝播速度と承認速度を測定するためのシミュレータを作成し，性能効率性とスケーラビリティをシミュレーションにより測定し，性能の比較を行った．そして，性能比較に基づき，IoT 分野に適用する際に適切であると考えられる分散台帳技術の技術特徴の分析を行った．本研究の貢献は以下の 2 点である．

1. IoT 環境を考慮した分散台帳技術の評価指標の提示
   IoT 環境の特性を考慮した上で分散台帳技術を評価する評価指標を提案した．提案する指標は性能効率性，スケーラビリティを重点においた指標である．指標は，構造的特徴を捉える一般的な性質と，シミュレーションにより測定する定量的な指標からなる．一般的な性質として，ネットワーク環境，ブロックサイズ，トランザクションサイズ，コンセンサス方式，ブロック生成時間の 5 点の項目を提示した．また，シミュレーションにより測定できる指標として，速度，スケーラビリティ，処理達成率を提示した．速度は，トランザクションの伝播速度，承認速度の 2 つとして定義した．

2. IoT 環境における分散台帳技術の性能評価手法の提案と分析
   提案した評価指標に基づき，Bitcoin の Blockchain，Ethereum，IOTA の 3 つの分散台帳について一般的な性質の比較を行った．これらの分散台帳は異なる特徴を有しており，ブロック生成時間，承認アルゴリズムの違いなどで明確な差が見られた．また，Ethereum についてシミュレータを構築し，シミュレーションを行った．これらの性能比較を通じて，分散台帳技術を IoT 分野に適用する際に適切である技術特徴について考察した．現状の分散台帳技術では，ノード規模の拡大や，それに伴うトランザクション数の増加に適切に対処できるとはいえないと

いう知見を得た．IoT 環境のようなトランザクションレートが高いと想定され，かつノード，トランザクションの規模が拡大していく環境では，一定時間に処理できるトランザクション数に上限のあるアルゴリズムを用いることは困難であると考える．

謝辞

本研究を行うにあたり，熱心なご指導，ご助言を賜りました林冬惠特定准教授に厚くお礼申し上げます．また，ご多忙にもかかわらず，快くアドバイザをお引受け下さるとともに，所属研究室では得られない視座からの大変有益なご助言をご提示くださった，京都大学大学院情報学研究科の加藤誠特定講師，現立命館大学情報理工学部の村上陽平准教授に対し深甚の謝意を表します．また，シミュレーションの構築にあたってご協力いただいた京都情報大学院大学の中口孝雄准教授に深く感謝申し上げます．そして，日頃より有益な御助言を与えてくださりました石田亨教授に深く感謝申し上げます．最後に，日頃から様々な御助言．御協力を頂きました石田・松原研究室の皆様に心から感謝の意を表します．

文献データベース

[Atzori et al., 2010] Atzori, L., Iera, A., and Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks*, 54(15):2787–2805.

[Bouloukakis et al., 2016] Bouloukakis, G., Georgantas, N., Billet, B., Bouloukakis, G., Georgantas, N., and Revisiting, B. B. (2016). Revisiting Service-oriented Architecture for the IoT : A Middleware Perspective To cite this version : Revisiting Service-oriented Architecture for the IoT : A Middleware Perspective.

[Croman et al., 2016] Croman, K., Decker, C., Eyal, I., Gencer, A. E., Juels, A., Kosba, A., Miller, A., Saxena, P., Shi, E., Sirer, E. G., et al. (2016). On scaling decentralized blockchains. In *International Conference on Financial Cryptography and Data Security*, pages 106–125. Springer.

[Decker and Wattenhofer, 2013] Decker, C. and Wattenhofer, R. (2013). Information propagation in the bitcoin network. In *Peer-to-Peer Computing (P2P), 2013 IEEE Thirteenth International Conference on*, pages 1–10. IEEE.

[Dorri et al., 2017a] Dorri, A., Kanhere, S. S., and Jurdak, R. (2017a). Towards an optimized blockchain for iot. In *Proceedings of the Second International Conference on Internet-of-Things Design and Implementation*, pages 173–178. ACM.

[Dorri et al., 2017b] Dorri, A., Kanhere, S. S., Jurdak, R., and Gauravaram, P. (2017b). Blockchain for iot security and privacy: The case study of a smart home. In *Pervasive Computing and Communications Workshops (PerCom Workshops), 2017 IEEE International Conference on*, pages 618–623. IEEE.

[Gubbi et al., 2013a] Gubbi, J., Buyya, R., Marusic, S., and Palaniswami, M. (2013a). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7):1645–1660.

[Gubbi et al., 2013b] Gubbi, J., Buyya, R., Marusic, S., and Palaniswami, M. (2013b). Internet of things (iot): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7):1645–1660.

[Kusmierz, 2017] Kusmierz, B. (2017). The first glance at the simulation of the tangle: discrete model.

[Nakamoto, 2008] Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system.

[Popov, 2016] Popov, S. (2016). The tangle. *cit. on*, page 131.

[Szabo, 1994] Szabo, N. (1994). Smart contracts. *Unpublished manuscript*.

[Szabo, 1997] Szabo, N. (1997). The idea of smart contracts. *Nick Szabo' s Papers and Concise Tutorials*, 6.

[Vermesan et al., 2009] Vermesan, O., Friess, P., Guillemin, P., Gusmeroli, S., Sundmaeker, H., Bassi, A., Jubert, I. S., Mazura, M., Harrison, M., Eisenhauer, M., Doody, P., Peter, F., Patrick, G., Sergio, G., Harald, Sundmaeker Alessandro, B., Ignacio Soler, J., Margaretha, M., Mark, H., Markus, E., and Pat, D. (2009). Internet of Things Strategic Research Roadmap. *Internet of Things Strategic Research Roadmap*, pages 9–52.