

## Blockchain のための遅延効率の良いピア選択手法

越川 幸輝<sup>†</sup> 蘇 悦<sup>†</sup> Jong-Deok Kim<sup>‡</sup> Won-Joo Hwang<sup>‡</sup> ゲエン キエン<sup>†</sup> 関屋 大雄<sup>†</sup>

Koshikawa Koki Yue Su Jong-Deok Kim Won-Joo Hwang Nguyen Kien Sekiya Hiroo

## 1. あらまし

ブロックチェーンは、IoT (Internet of Things)のセキュリティなどの課題を解決する技術として期待されている。しかし、それらを組み合わせた IoT-Blockchain は低い拡張性が問題となる。この拡張性は、システム上のノードが直接接続、通信するピアに依存する。しかし、現在最も普及しているピア選択は、遅延を全く考慮せず、ランダムである。そこで遅延を考慮し、低減するピア選択を実装する必要がある。本研究では、Transaction-oriented latency(TOL)とBlock-oriented latency(BOL)を考慮し最適化する Dual Perigee を提案する。Dual Perigee の有効性を示すため、IoT-Blockchain システムをエミュレートし、従来のピア選択と比較した。

## 2. はじめに

近年、IoT は多くの分野で応用されている[1]。しかし、IoT は多様なネットワークプロトコルやデバイスを使用しているため、プライバシーやセキュリティに関わる相互運用性の問題を抱えている。IoT の相互運用性に関連する問題をブロックチェーン技術によって、解決できる可能性がある。ブロックチェーン上のデータはその完全性や透明性、追跡性により高い信頼性を実現する。さらに、コンセンサスアルゴリズムにより、第三者の機関を介することなく、ネットワーク全体から合意を得ることができる。この合意形成を通して生まれる非中央集権制により障害に強いシステムとなる。また、これらの利点に加え、ブロックチェーンは IoT システムの多様なネットワークデバイスやプロトコルを抽象化する、ミドルウェアとしての役割も果たす[2]。この抽象化により、IoT アプリケーションの実装を容易化する。結果として、IoT とブロックチェーン技術を組み合わせた技術、IoT-Blockchain が注目を浴びている。

Blockchain システムには多くの利点があるが、拡張性が低くなってしまふ問題がある[3]。この拡張性は通常 1 秒あたりに処理されるトランザクションの処理数を表し、スループットで評価される。スループットはコンセンサスアルゴリズム、検証時間、ネットワーク遅延に依存する。検証時間は、PoA(Proof-of-Authority)のような軽量なコンセンサスメカニズムを使用することで、最小化することができる。したがって、Blockchain の拡張性を向上させるために考慮すべきことが、P2P ネットワーク上におけるメッセージの伝搬遅延である[4]。ブロックチェーン上の長い伝搬遅延の主な原因は、P2P ネットワーク上のノード間が無作為に接

続され、形成されるオーバーレイネットワークである。例えば、東京にあるノード A、B、大阪にあるノード C を考える。この時、無作為に形成されたオーバーレイネットワークではデータの送信が A、C、B の順で行われる可能性がある。東京間のノードでデータを同期するときでさえ、一度大阪を介するのは明らかに高遅延の原因となる。一度遠く離れたノードを介するのは遅延の観点から非効率であるため、IoT-Blockchain を実現するには効率的なオーバーレイネットワークを設計することが不可欠である[5,6]。

文献[7]では、著者らにより Perigee アルゴリズムと呼ばれるピア選択アルゴリズムが提案された。Perigee アルゴリズムでは、過去のトランザクションの遅延時間を計測し、遅延時間が短いピアとの接続を維持し、長いピアを他のノードと入れ替える。この入れ替えにより、Perigee アルゴリズムはイーサリアムのデフォルトのピア選択方法より、トランザクションの伝搬遅延を低減する。しかし文献[5]にある通り、トランザクションライフサイクルで定義されているブロック伝搬遅延が Perigee アルゴリズムでは全く考慮されていない。本研究では、プライベートイーサリアム環境の IoT-Blockchain システムにおいて、ピア選択アルゴリズムである Dual Perigee アルゴリズムを提案する。トランザクションの遅延時間のみを計測する Perigee アルゴリズムとは異なり、Dual Perigee アルゴリズムではトランザクションの遅延時間とブロックの遅延時間を計測する。そして、この計測された遅延時間を入力とするスコア関数を定義した。この新たなスコア関数の導入によりトランザクション、ブロックの合計遅延が小さいピアとの接続を維持することが可能である。Dual Perigee アルゴリズムの有効性を示すため、100 ノードのエミュレートされた IoT-Blockchain システム上でデフォルトのピア選択、Perigee、Dual Perigee アルゴリズムを比較した。その結果、Dual Perigee アルゴリズムは Perigee アルゴリズムよりブロック伝搬遅延の低減に優れていることが示された。さらに、Dual Perigee アルゴリズムの示した遅延時間はデフォルトのピア選択の 43%、Perigee アルゴリズムの 80%であった。

## 3. 研究背景と関連研究

## 3.1 IoT-Blockchain

ブロックチェーンは高いセキュリティ、プライバシー、非中央集権制により IoT のパフォーマンスを向上させる。Kashetri[8]らはブロックチェーンの非中央集権制により認証システム、アクセス管理の性能が向上し、障害に強いシステムが実現すると述べている。それに応じて IoT-Blockchain を実現する動きが盛んになっている。文献[2]の著者は IoT-Blockchain のレイヤーアーキテクチャを提案した。IoT-Blockchain は、多くのデバイスやネットワーク上に構築される。このネットワークの多様性は、潤滑なアプリケーション開発の妨げとなる。しかし、アプリケーション開発者はブロックチェーンのスマートコントラクトを

<sup>†</sup> 千葉大学大学院 Graduate School of Science and Engineering, Chiba University, 1-33, Yayoi-cho, Inage-ku, Chiba-shi, Chiba, 263-8522 Japan

<sup>‡</sup> 釜山大学 School of Computer Science and Engineering, Pusan National University, 2 Busandaehak-ro 63beon-gil, Geumjeong-gu, Busan, South Korea  
Corresponding author: nguyen@chiba-u.jp

API として使用することで、サービスの実装が容易になる。その結果、広い分野で IoT-Blockchain の応用が期待されている。文献[9]では Blockchain が実装される分野が提示されている。ヘルスケアシステムは中でも有力な応用分野である。患者の医療データは高いプライバシー基準を要する。さらに、医療ミスが減らすためにもデータの信頼性は欠かせない要素である。Blockchain はこれらの懸念事項を解決する手段となる。さらに、自動運転技術もブロックチェーンによる耐障害性により、応用先として期待されている。しかし、高まる IoT の需要の中で、ブロックチェーンの低い拡張性は対処されるべき懸念事項である。ブロックチェーンの低い拡張性の 1 つの要因は、P2P ネットワーク上のノード間の非効率な接続方法が起因する長いネットワーク伝搬遅延である。

### 3.2 ブロックチェーンピア選択

イーサリアムは DEVp2p と呼ばれるプロトコル上に実装される。このプロトコルは、ノード間の接続を管理するために使用されるノードディスカバリプロトコルと RLPx トランスポートプロトコルで構成される。ノードディスカバリプロトコルでは、Kademlia を改良した分散ハッシュテーブル(DHT)を使い、ノードの情報を共有する。そして、それぞれのノードは近いピアと接続する。この近さとは地理的なものではなく、それぞれのノードが持つ 256 ビットのノード ID の排他的論理和で定義される。この 256 ビットの ID は Secp256k1 楕円曲線から無作為に生成される。そして、接続されたノード同士は互いにノードの情報を要求しあい、収集したデータを DHT に保存する。RLPx はイーサリアムノード同士が通信するための、TCP を用いたトランスポートプロトコルである。RLPx では Recursive Length Prefix(RLP)を用いて送信データをエンコードし、圧縮する。RLPx セッションは暗号化ハンドシェイクから始まり、ユーザーに安全な通信路を提供する。これらのプロトコルは無作為に生成された ID に依存しており、ランダムで非効率な P2P ネットワークを形成する。しかし、これはイーサリアムクライアントソフト Geth のデフォルトの設定である。

文献[10]では、著者らが IP アドレスから地理的な位置を推測し、近いノード同士を接続する方法を提案した。地理的に近いノード同士を接続することはメッセージの伝搬遅延低減に繋がり、効率的な P2P ネットワークを形成する。しかし、この手法は VPN やプロキシサーバーを使用した際に、地理的な位置の推測が困難になってしまう問題がある。文献[7]では、低遅延のピアと状況に応じ接続する Perigee アルゴリズムが提案されている。このアルゴリズムはリンクごとの遅延や詳細な地理的情報を使用しないことから軽量である。加えて、遅延を実際に計測することから、自動的にリンクごとの遅延や帯域幅、処理遅延も考慮されることとなる。しかし、トランザクションの伝搬遅延である TOL(Transaction-oriented latency)とブロックの伝搬遅延である BOL(Block-oriented latency)[5]を考慮したときに Perigee は TOL しか考慮していないことがわかる。そして、このアルゴリズムは BOL の削減を考慮しない。したがって、ピア選択で BOL を考慮に入れる必要がある。

## 4. Dual Perigee: 遅延最小化のためのピア選択

本章では最初に、基準となるピア選択アルゴリズムである Perigee アルゴリズム[7]について述べる。このアルゴリズムでは現在のピアの情報をもとにしてピアを選択し、direct global latency[12]を低減する。そして、私たちの提案アルゴリズムである Dual Perigee アルゴリズムの説明を行う。

### 4.1 Perigee と direct global latency

1) Direct global latency: Perigee アルゴリズムではブロックチェーンの P2P ネットワークを  $\mathcal{N} = (\mathcal{V}, \mathcal{E})$  としてモデル化する。それぞれのノード  $v \in \mathcal{V}$  はトランザクション  $T_x$  を作成し、ネットワーク全体にブロードキャストする。 $T_x$  はノード  $v$  に送信され、受信したノードはすぐに  $T_x$  を他のノードへと送信する。ここで  $\Lambda_a(T_x)$  をあるランダムなソースノードからあるノード  $a$  までのトランザクションの遅延とする。このときノード  $a$  に関する direct global latency を以下のように定義した。

$$L_{T_x}(a) = \mathbb{E}[\Lambda_a(T_x)] \quad (1)$$

ノード  $a$  に関する direct global latency とはノード  $a$  に届くトランザクションの遅延をランダムなソースノードに関して平均したものである。また、ノードが維持できるピアの最大数を  $k$  とすると、direct global latency の最適化問題は以下のようにあらわされる。

$$\begin{aligned} & \text{Minimize } L_{T_x}(a) \\ & \text{Subject to } |\{y | (a, y) \in \mathcal{E}\}| \leq k \end{aligned} \quad (2)$$

この最適化問題は Single-source shortest path distance minimization と呼ばれる NP 困難な最小化問題である。つまり、たとえ P2P ネットワーク  $\mathcal{N}$  のすべての情報を持っていたとしても、式(2)を現実的な時間内に解くのは不可能である。さらに、現実のネットワークはリンクごとに帯域幅、遅延時間が異なるため、式(2)を満たすピアを選択するのは困難である。

2) Perigee アルゴリズム: Perigee アルゴリズム[12]はノード間の通信状態に基き、式(2)に近い状態に収束するための分散アルゴリズムである。Perigee アルゴリズムはまず、無作為に選択したピアと接続し、それぞれのピアを評価する。そしてトランザクションの伝搬遅延が小さくなるピアを選択する。低遅延のノードをピアとして維持し、それ以外は新しく無作為に選択したノードと入れ替える。この入れ替えの後、Perigee アルゴリズムは direct global latency を最小化する最適なトポロジーに近い状態に収束する。

Perigee アルゴリズムはトランザクションライフサイクル[5]の 1 部分であるトランザクション伝搬遅延の最適化のみを考慮する。つまり、ブロックの伝搬遅延低減について全く考慮されていない。この問題に対し、Perigee アルゴリズムを改良することで、トランザクション、ブロック両者の伝搬遅延時間である TOL と BOL を考慮し、低減する Dual Perigee アルゴリズムを提案する。

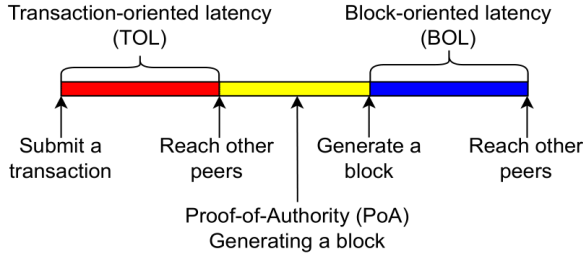


図1 TOLとBOL

## アルゴリズム 1: Dual Perigee アルゴリズム

入力: ネットワーク上のノード:  $\mathcal{V}$ , 最大ピア数:  $k$ ,インターバル:  $t$ , 置換率:  $r$ 出力: 低遅延のピアの集合:  $P^*$ 制約:  $1 \leq k \leq |\mathcal{V}|, 0 < r < 1$  $P \leftarrow \text{randomPeerSelection}(\mathcal{V}, k);$ for  $epoch = 1, 2, \dots$  doSleep( $t$ ); $\Phi \leftarrow \text{Init Score Map};$ for  $p \in P$  do     $\Phi(v) \leftarrow \phi(v);$      $P \leftarrow P - \text{least}(\Phi, r);$      $P \leftarrow P \cup \text{randomPeerSelection}(\mathcal{V}, r);$ 

## 4.2 Dual Perigee

1) Transaction-oriented latency と Block-oriented latency: トランザクションライフサイクルでは送信ノードからトランザクションが送信され、他のピアへと伝搬する。トランザクションを受信したブロック生成ノードはそれらを1つのブロックへまとめ、ネットワーク全体へと送信する。文献[5]では、トランザクションが送信されてから他のノードに受信されるまでの遅延を TOL、ブロックが生成されてから他のノードが受信するまでの遅延を BOL として定義されている。この TOL と BOL の概念図を図1に示した。図1における赤と青で示された部分がそれぞれ TOL と BOL に対応する。また、黄色で示された部分はトランザクションをまとめ、ブロックを生成するための処理遅延に対応する。処理遅延を除いた TOL と BOL の合計遅延はブロックチェーン上のネットワーク遅延全体を表す。この合計遅延を低減することがブロックチェーンの拡張性を向上させるのにつながる。

Perigee アルゴリズム同様、遅延について最小化問題を以下のように定義した。トランザクション  $Tx$  について  $A_a(Tx)$  をソースノードからノード  $a$  への TOL と定義した。このとき  $L_{Tx}(a) = \mathbb{E}[A_a(Tx)]$  とする。さらにブロック  $Blk$  について  $A_a(Blk)$  をソースノードからノード  $a$  への BOL と定義した。また、 $L_{Blk}(a) = \mathbb{E}[A_a(Blk)]$  とする。TOL、BOL に関する direct global latency を組み合わせた最適化問題を以下式(3)のように定義した。

$$\begin{aligned} & \text{Minimize } L_{Tx}(a) + L_{Blk}(a) \\ & \text{Subject to } |\{y | (a, y) \in \mathcal{E}\}| \leq k \end{aligned} \quad (3)$$

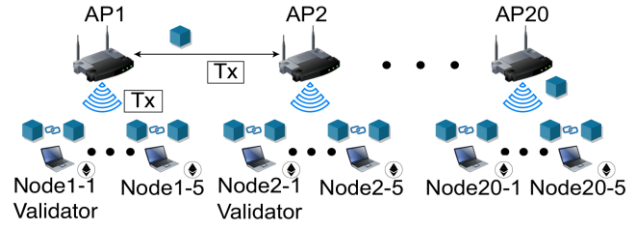


図2 エミュレートされた IoT-Blockchain システム

式(3)を満たすことで、ブロックチェーン上の複数の送信ノードからノード  $a$  までの TOL と BOL を減らすことが可能である。

## 2) Dual Perigee アルゴリズム

式(3)を達成するための Dual Perigee アルゴリズムをアルゴリズム1に示す Dual Perigee アルゴリズムはネットワークのノード  $\mathcal{V}$ 、最大のピア数  $k$ 、インターバル  $t$ 、置き換え比率  $r$  を入力として受け取る。Perigee アルゴリズムと同様、Dual Perigee アルゴリズムは最初、無作為に  $k$  個のピアを選択し、それぞれのピアにスコアを割り当てる。そして、Dual Perigee アルゴリズムでは  $t$  秒ごとに遅延の大きい  $k \times r$  個のピアを、他の無作為に選択したノードと入れ替える。このとき低遅延だった残りのピアはそのまま接続が維持される。この入れ替えを繰り返し、時間とともに最適なピアに収束する。Dual Perigee アルゴリズムでは TOL と BOL の両者を低減するためのスコア関数を実装した。 $u$  をネットワーク上のあるノードとする。あるインターバルについて、 $M_{Tx}^v$  を現在のピア  $v$  からノード  $u$  に送信されたトランザクションの集合とする。また、 $M_{Blk}^v$  を現在のピア  $v$  からノード  $u$  に送信されたブロックの集合とする。トランザクション  $Tx$  について、任意のピアからノード  $u$  が最初にトランザクション  $Tx$  を受信した時刻を  $T(Tx)$ 、ピア  $v$  から受信した時刻を  $T_v(Tx)$  とする。このとき、ピア  $v$  からトランザクション  $Tx$  が受信できなかった場合は、 $T_v(Tx) = \infty$  となる。トランザクション同様、ブロックについても受信時間を定義した。任意のピアからノード  $u$  が最初にブロック  $Blk$  を受信した時刻を  $T(Blk)$ 、ピア  $v$  から受信した時刻を  $T_v(Blk)$  とする。そして、Dual Perigee アルゴリズムのスコア関数を以下のようにして定義した。

$$\begin{aligned} \phi(v) = & \sum_{Tx \in M_{Tx}^v} \frac{1}{|M_{Tx}^v|} \min\{T_v(Tx) - T(Tx), \bar{\Delta}\} \\ & + \sum_{Blk \in M_{Blk}^v} \frac{1}{|M_{Blk}^v|} \min\{T_v(Blk) - T(Blk), \bar{\Delta}\} \end{aligned} \quad (4)$$

さらに、定数  $\bar{\Delta}$  を計測された遅延の差の上限として、定義した。これにより  $T_v(Tx) = \infty$  となった場合でもあってもスコア関数が発散しない。Dual Perigee アルゴリズムでは式(4)がスコア関数として使われる。Dual Perigee アルゴリズムではスコアが大きいピアを他のノードと入れ替えることで、できるだけ最小化されたスコアを目指す。小さいスコアを示したピアであるほどピアとして接続され続ける。低スコアは、そのピアから送信されるトランザクションやブロックの受信時刻が早かったことを示す。結果として、スコアの最小化により、TOL と BOL を低減させるピアを選択でき、ネットワーク遅延全体を縮小することにつながる。

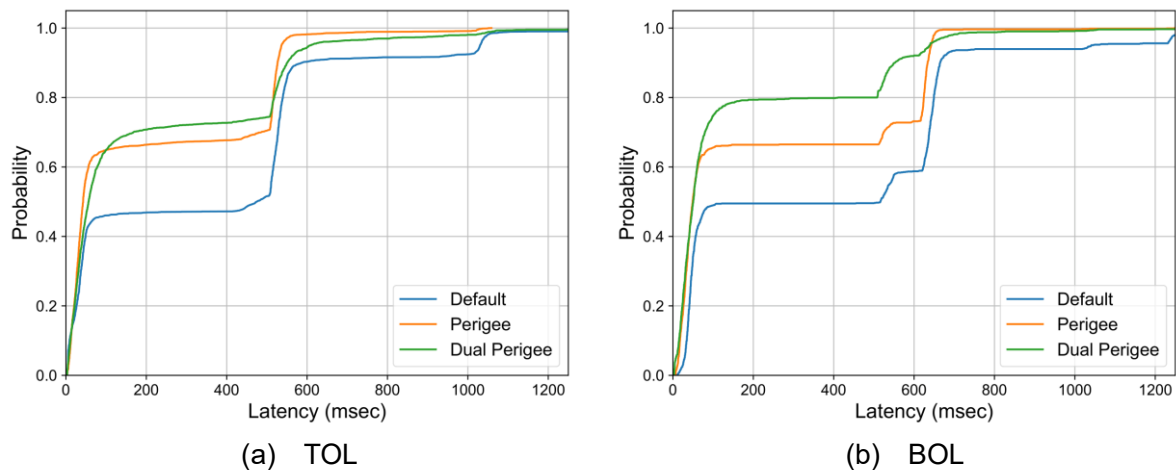


図 3 TOL と BOL の累積確率分布

## 5. 評価

### 5.1 評価設定

100 ノードで構成される IoT-Blockchain システムをエミュレートするため、Mininet-Wifi[13]を使用した。Mininet-Wifi は Linux 無線デバイスドライバを使用したエミュレーターソフトである。このソフトはブロックチェーンのようなアプリケーションの実装を可能にする、軽量な仮想化技術を提供する。最初に、20 個のアクセスポイント(AP)を直線状に有線で接続した。それぞれの AP は 5 個のノードと wifi で接続した。作成したシステムを図 2 に示した。それぞれのノード上に Geth[14]というイーサリアムクライアントソフトを起動した。そしてリソースが制限された IoT 環境で適した、Proof-of-Authority(PoA)コンセンサスアルゴリズムを使用したプライベートイーサリアムブロックチェーンを構築した。AP1 から AP5 までの 5 個の AP にそれぞれ 1 つずつブロック生成ノードを指定した。4 秒間隔で無作為に選ばれたノードがブロックを生成し、送信する。加えて、IoT 環境をシミュレートするためのスマートコントラクトをイーサリアムブロックチェーン上にデプロイした。このスマートコントラクトは 2 つの機能を持つ。1) ブロックチェーン上に値を書き込む。2) ブロックチェーン上の現在の値を読み取る。スマートコントラクトを実行する間に出力されるログのうち、TOL と BOL の計算に必要なものを収集した。このスマートコントラクトの実行にはトランザクションを生成する必要がある。本実験ではすべての AP にトランザクションを送信するノードを接続し、それらのノードは 10 秒間隔でトランザクションを送信する。この実験をブロックが 100 個生成されるまでの期間で行った。

提案アルゴリズムの評価のため、ノード 20-5 には 3 種類のイーサリアムクライアントを実装した。デフォルトの設定でピア選択を行う通常の Geth(Default)、Perigee アルゴリズムを実装した Geth(Perigee)、そして Dual Perigee アルゴリズムを実装した Geth(Dual Perigee)である。すべての Geth について最大のピア数を 5 に設定した。Perigee、Dual Perigee アルゴリズムに関しては、ピアの入れ替えインターバル、入れ替え比率をそれぞれ 120 秒、0.2 にした。つまり、

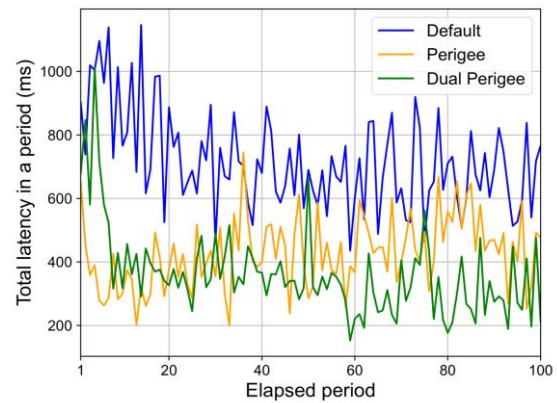


図 4 Total latency in a period の時間変化

Perigee、Dual Perigee アルゴリズムを実装した際、120 秒ごとに 5 個のピアから 1 つのピアを選択し、無作為に選ばれたピアと入れ替える。それぞれの実験において、20 個のトランザクション送信ノードからノード 20-5 までの TOL、5 個のブロック生成ノードからノード 20-5 までの BOL を計測した。

### 5.2 結果

収集した TOL と BOL の累積確率分布をそれぞれ図 3(a)、図 3(b)に示した。両者の累積確率分布は階段状になっており、遅延が大きいものと小さいものが極端に存在していることを示していた。この遅延の差はたとえ、同じソース元からのトランザクションやブロックだとしても観測された。この原因を調べるため、パケットキャプチャツールで AP に流れるパケットを観測したところ、TCP 再送信が観測された。これは、イーサリアム上の通信が TCP を使用した RLPx トランスポートプロトコル上で通信が行われるためである。つまり、遅延の大きいトランザクションの原因は TCP 再送信であることがわかった。この TCP 再送信は割り当てられた時間内にパケットを受信できなかった場合や、パケットロスが起こった際に発生する。

図 3(a)は Perigee、Dual Perigee アルゴリズムを用いることによって、TOL が低減されていることを示している。これ

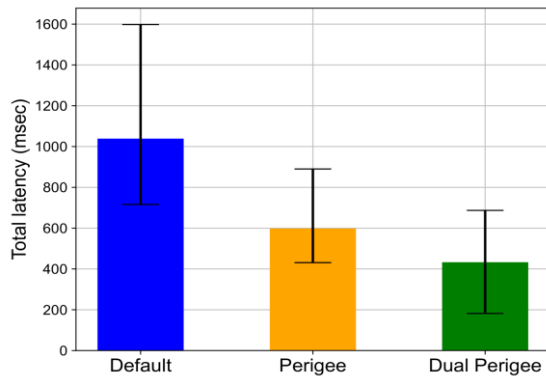


図 4 10 回の実験における Total latency

はノード 20-5 が長い遅延やパケットロスが起きやすいピアを避け、TOL の短いピアを選択したことを示す。そして、図 3(b)は BOL の削減に最も効果的だったのは Dual Perigee アルゴリズムであったことを示している。この結果は Dual Perigee アルゴリズムのスコア関数においてブロック受信時刻が使われ、BOL が低減されたことを意味する。これら図 3 の結果により、Dual Perigee アルゴリズムは TOL、BOL の両者を減らすことに効果的であることが示された。

次に時間変化に対する遅延の変化を比較するため、実験期間を 100 の期間に分割した。それぞれの期間で受信したトランザクションやブロックに関し、20 個のトランザクション送信ノードからノード 20-5 までの平均 TOL、5 個のブロック生成ノードからノード 20-5 までの平均 BOL を計算した。そして、これら 2 つの平均遅延の合計を Total latency in a period として導出した。この Total latency in a period はそれぞれの期間における、ブロックチェーン上のネットワーク遅延全体を示している。この Total latency in a period を図 4 に示した。実験開始付近は全てのアルゴリズムで類似した Total latency in a period が観測された。しかし、Perigee、Dual Perigee アルゴリズムでは動的にピアを選択することにより、時間が進むにつれ遅延が低減された。また、Dual Perigee アルゴリズムは Perigee アルゴリズムより BOL の低減に効果的であるため、高い削減率を示した。

最後に、無作為にピアを選択することによるピアの初期状態によらない一般性を示すため、前述した実験を 10 回繰り返した。まずすべての実験において、20 個のトランザクション送信ノードからノード 20-5 に届くときの平均 TOL、5 個のブロック生成ノードからノード 20-5 までの平均 BOL を計算した。そして実験ごとに導出される平均 TOL と平均 BOL の合計である Total latency を 10 回分記録した。そのときの最大値、平均値、最小値を図 5 に示した。

実験ごとに 10 回計測された Total latency の平均を比較すると、Dual Perigee アルゴリズムの遅延はデフォルトのピア選択の Geth における遅延の 43% 程度になった。さらに Perigee アルゴリズムと比較すると、Dual Perigee アルゴリズムの遅延は 80% 程度になった。この差はスコア関数にブロック受信時刻を組み合わせることで、BOL が効果的に削減されたことを示している。

## 6. 結論

本研究では、ブロックチェーン上のトランザクションライフサイクルを考慮することにより、遅延時間を短縮化する P2P ネットワークを形成する Dual Perigee アルゴリズム

を提案した。Dual Perigee アルゴリズムではスコア関数として TOL と BOL の両者を考慮に入れた。このスコア関数により、低遅延のピアとの接続を維持することが可能になった。そして、Geth を改良することで提案アルゴリズムを実装し、100 ノードのエミュレートされた IoT-Blockchain 上で評価した。

そして、Dual Perigee アルゴリズムを最新の Perigee アルゴリズム、Geth のデフォルトのピア選択アルゴリズムと比較した。その結果、Dual Perigee アルゴリズムは他の 2 つのアルゴリズムより優れた遅延性能を示した。

## 謝辞

本研究は JSPS 科研費 JP20H0417, JST 科学技術イノベーション創出に向けた大学フェロウシップ創設事業 JPMJFS2107 の支援を受けたものです。

## 参考文献

- [1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015
- [2] H.-N. Dai, Z. Zheng, and Y. Zhang, "Blockchain for internet of things: A survey," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8076–8094, 2019
- [3] M. A. Uddin, A. Stranieri, I. Gondal, and V. Balasubramanian, "A survey on the adoption of blockchain in iot: challenges and solutions," *Blockchain: Research and Applications*, vol. 2, no. 2, p. 100006, 2021.
- [4] X. Wang, X. Zha, W. Ni, R. P. Liu, Y. J. Guo, X. Niu, and K. Zheng, "Survey on blockchain for internet of things," *Computer Communications*, vol. 136, pp. 10–29, 2019
- [5] X. Chen, K. Nguyen, and H. Sekiya, "On the latency performance in private blockchain networks," *IEEE Internet of Things Journal*, vol. 9, no. 19, pp. 19246–19259, 2022
- [6] Y. Su, K. Nguyen, and H. Sekiya, "A comparison of blockchain recovery time in static and mobile iot-blockchain networks," *Future Internet*, vol. 14, no. 11, 2022.
- [7] Y. Mao, S. Deb, S. B. Venkatakrishnan, S. Kannan, and K. Srinivasan, "Perigee: Efficient peer-to-peer network design for blockchains," 2020.
- [8] N. Kshetri, "Can blockchain strengthen the internet of things?," *IT Professional*, vol. 19, no. 4, pp. 68–72, 2017.
- [9] S. Mathur, A. Kalla, G. Gür, M. K. Bohra, and M. Liyanage, "A survey on role of blockchain for iot: Applications and technical aspects," *Computer Networks*, vol. 227, p. 109726, 2023.
- [10] O. Abboud, A. Kovacevic, K. Graffi, K. Pussep, and R. Steinmetz, "Underlay awareness in p2p systems: Techniques and challenges," in *2009 IEEE International Symposium on Parallel Distributed Processing*, pp. 1–8, 2009.
- [11] A. Meyerson and B. Tagiku, "Minimizing average shortest path distances via shortcut edge addition," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques* (I. Dinur, K. Jansen, J. Naor, and J. Rolim, eds.), (Berlin, Heidelberg), pp. 272–285, Springer Berlin Heidelberg, 2009.
- [12] W. Tang, L. Kiffer, G. Fanti, and A. Juels, "Strategic latency reduction in blockchain peer-to-peer networks," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 7, p. 1–33, May 2023
- [13] R. R. Fontes, S. Afzal, S. H. B. Brito, M. A. S. Santos, and C. E. Rothenberg, "Mininet-wifi: Emulating software-defined wireless networks," in *2015 11th International Conference on Network and Service Management (CNSM)*, pp. 384–389, 2015.
- [14] "Ethereum geth client." <https://geth.ethereum.org/>. Access Date: 2024 January.
- [15] J. Abdella, Z. Tari, A. Anwar, A. Mahmood, and F. Han, "An architecture and performance evaluation of blockchain-based peer-to-peer energy trading," *IEEE Transactions on Smart Grid*, vol. 12, no. 4, pp. 3364–3378, 2021.