

IoT の IOTA を用いた Micropayment 方式

IOTA-based Micropayment of IoT Data

屋敷 圭太

上山 憲昭

Keita Yashiki

Noriaki Kamiyama

立命館大学 情報理工学部 情報理工学科

College of information Science and Engineering, Ritsumeikan University

1. はじめに

IoT では様々なプラットフォーム、デバイス所有者が相互にデータを共有、提案することで豊富なサービスを利用することを可能にしている。IoT ではサービスの享受に伴い、頻繁にサービス利用者とデバイス所有者、サービスプラットフォーム間で支払いが発生する。アプリケーションの支払いシステムとして、少額電子決済である micropayment が注目されている。micropayment は、数円や数十円ほどのわずかな金額を扱う電子決済を指す。この種の支払いは、IoT デバイスなどで少量のデータを扱う場合に利用されることが期待されている。しかし二重支払いを防ぐための仕組みやスケーラビリティ向上のための処理の分散化が必要である。また単一の事業者が全ての支払い処理を行うとプライバシー、スケーラビリティ、セキュリティの観点で問題が発生する。これらの問題を解決するため、IoT データの micropayment の決済方法として、分散台帳技術を採用した仮想通貨の使用が注目されている。

本稿では buyer と seller 間での騒音データの取引を想定し、細かい金額の粒度で支払いを行うことが出来る micropayment による取引に際し、取引手数料が無料で高いスケーラビリティをもつ分散台帳技術の一つである IOTA 上で取引を行うことで、支払いをまとめて行うよりも処理が軽く遅延時間が短い IoT 取引が可能な方式を提案する。通常、少額の支払いを行う場合、手数料が相対的に大きな割合を占めることやスケーラビリティが問題となるが、micropayment に IOTA を用いることで手数料無料でスケーラビリティの問題が解消される。分散型台帳としては Blockchain が代表的であるが、Blockchain はスケール性に課題があるのに対し、IOTA はスケール性が高い。提案方式では、支払いの粒度を細かくすることを想定し、transaction 数が 100, 500, 1,000 の各場合における台帳内でコンテンツ名を検索する四つの探索手法の検索時間と、必要メモリ量を比較する。そして検索時間と必要メモリ量のトレードオフの関係を確認する。

2. IOTA

IOTA では、新しいトランザクションが未承認のトランザクションである「tip」から 2 つを選び、それによって非巡回グラフ (DAG: directed acyclic graph) である「Tangle」が形成され、分散型台帳が構築される。新しく DAG に追加されたトランザクションは、それぞれ既存の 2 つのトランザクションを参照する。また、各トランザクションには重みがあり、直接的および間接的に承認した各トランザクションの重みが、その承認されたトランザクションの累積重みとして表される。また、選択されていない先頭にあるトランザクションは「tip」と呼ばれる。Tip 選択アルゴリズムには、tip からランダムに 2 つを選択する URS (uniform random selection) および DAG 内の最初のトランザクションであるジェネシストランザクションから参照されているトランザクションを等確率で選択していく方法があり、これにより 2 回の選択で 2 つの tip を得る URW (unweighted random walk) および URW と同様にジェネシスノードから選択し、トランザクションの累積重みを考慮して選択する WRW (weighted random walk) の 3 つがある。WRW では、 H_x , H_y をトランザクション x および y の累積重み、 α (≥ 0) を累積重みのパラメータとすると、トランザクション y から x への遷移確率 P_{xy} は次式で定義される。

$$P_{xy} = \frac{e^{-\alpha(H_x - H_y)}}{\sum_{z: z \rightarrow x} e^{-\alpha(H_x - H_z)}} \quad (1)$$

累積重みのパラメータ $\alpha = 0$ のときは重みのないランダムウォーク、つまり URW となる。

3. 提案方式

seller は、コンテンツ名の prefix, ID, 公開鍵, デジタル署名, コンテンツ名を transaction に登録する。本稿では seller

が登録する騒音データをコンテンツとして扱う。また DAG の検索法として、ハッシュチェーン法、二分探索木 (bst: binary search tree), 幅優先探索 (bfs: breadth-first search), 深さ優先探索 (dfs: depth-first search) の四つを考える。

ハッシュチェーン法では、コンテンツの prefix を数値に変換し、それをハッシュ関数にかけて得られるハッシュ値を使用し、ハッシュテーブルの各バケットにデータを格納する。複数の異なる prefix が同じハッシュ値に変換される可能性があるため、ハッシュチェーン法ではこれらのデータを連結リストで結びつけ、同じバケットに複数のデータを管理する。提案方式では、ハッシュテーブルを使用してコンテンツの prefix と ID を管理し、その ID を元に DAG 上で直接アクセスしてコンテンツ名を取得する。二分探索木 (bst: binary search tree) も同様に、木構造でコンテンツの prefix と ID を管理する。prefix を数値に変換し、二分探索木で対応する ID を見つけてから、DAG 上で直接アクセスしてコンテンツ名を取得する。

幅優先探索 (bfs: breadth-first search) では、DAG 上のジェネシストランザクションからのホップ数が小さいトランザクションから順に探索する。同じホップ数のトランザクションを探索しても見つからない場合は、ホップ数を一つ増やして再度探索を行う。深さ優先探索 (dfs: depth-first search) も同様に、ジェネシストランザクションから探索を始め、子を持たないトランザクション (tip) に到達するまで深く伸びて探索する。tip に到達してもまだ見つからない場合は、最後に分岐した箇所まで戻り、未探索のトランザクションを探索する。ハッシュチェーン法と二分探索木は prefix と ID を管理しているため、DAG 上で直接アクセスしてコンテンツ名を取得する必要がある。一方で、幅優先探索と深さ優先探索では、トランザクションにコンテンツ名が管理され、発見した時点で探索が終了する。これらの手法を比較し、検索時間と必要メモリ量を評価する。

seller がコンテンツ名を登録する際は、システムは重複がないかを確認するために全探索を行い、既に存在していれば登録を拒否し、存在しなければ登録する。検索時間は平均値、中央値、および最大値の上位 95 % 値で評価し、必要メモリ量は幅優先探索および深さ優先探索を DAG としてまとめ、ハッシュチェーン法と二分探索木に対して比較する。

4. 性能評価

提案方式を計算機シミュレーションにより評価する。transaction 数 100 と 500 と 1,000 の三通りで、URS の tip 選択アルゴリズムで DAG を生成する。1 秒当たりの transaction 生成間隔は平均が 50 (/秒) の指数分布に従い発生させる。buyer による決済要求は、総要求数を 5,000 とし、発生間隔を平均 50 (/秒) の指数分布に従い発生させる。またハッシュチェーン法におけるテーブルのバケット数を 100 とする。

図 1 に該当 transaction の探索に要する検索時間の平均値、中央値、95 % 値を DAG 上のトランザクション数 T の 3 つの場合について各々示す。いずれの transaction 数の場合も検索時間はハッシュチェーン法 < 二分探索木 < 深さ優先探索、幅優先探索となる。transaction 数 100 と 500 の場合は深さ優先探索と幅優先探索の検索時間にほぼ差がないことを確認した。transaction 数 1,000 では transaction 数 100 に比べて、幅優先探索よりも深さ優先探索の方が探索時間が大きく、深さ優先探索、幅優先探索に対してハッシュチェーン法、二分探索木との検索時間の差がより大きくなる。

ハッシュチェーン法ではハッシュ値を計算してバケットにアクセスし、その中で管理されているリストを検索するが、 T が小さい場合はバケットで管理されているデータ数も少ないため、他の手法と比べて短い時間で全探索が可能である。二分探索木においても全てのデータではなく、必要な部分を探索するため検索時間も比較的短い。一方で深さ優先探索と幅優先探索では、浅いノード層から深いノード層までしらみ潰しに DAG を全探索するため時間を要する。

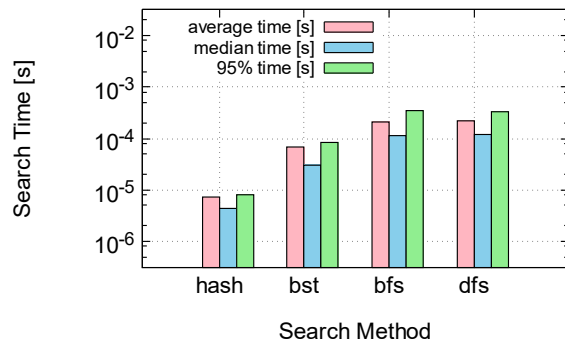
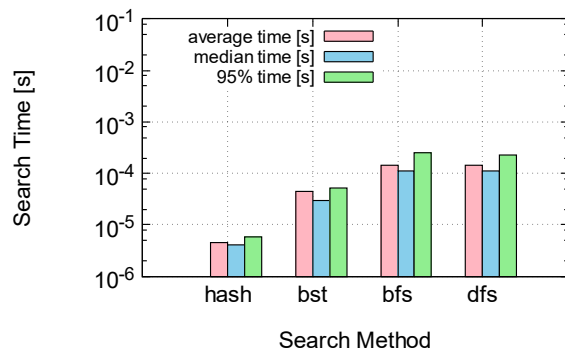
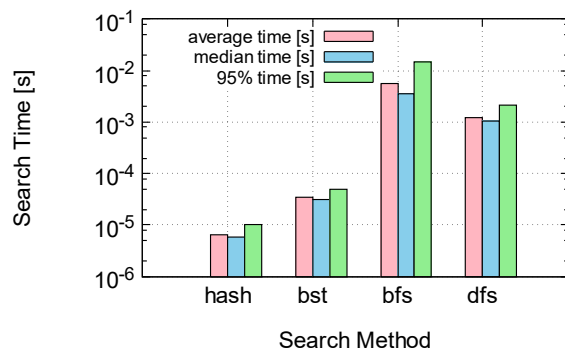
(a) $T = 100$ (b) $T = 500$ (c) $T = 1000$

図 1: Required search time

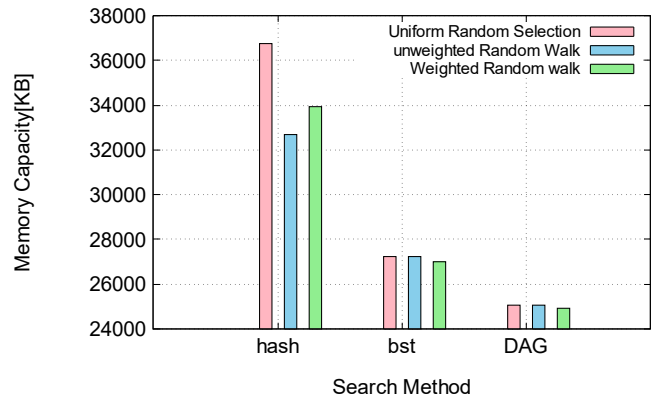
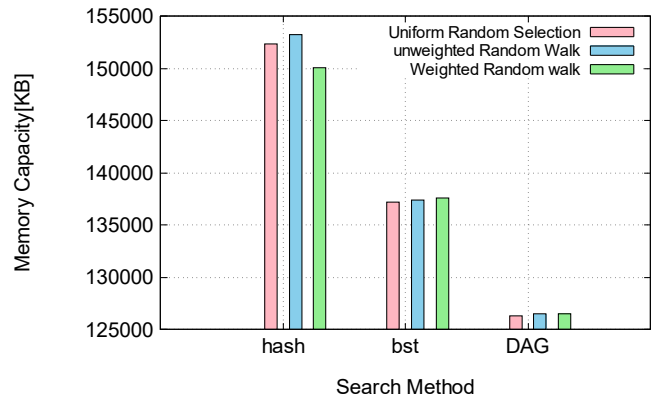
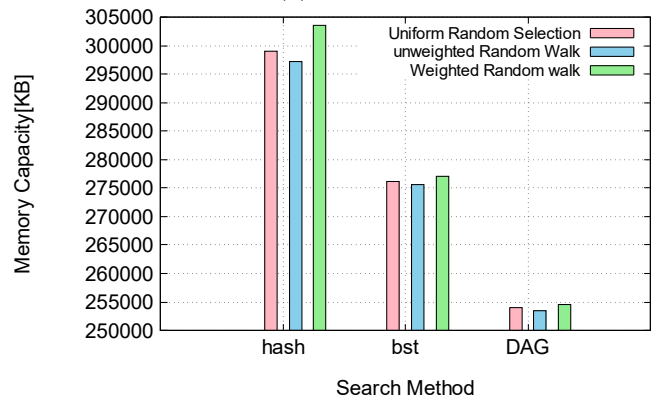
(a) $T = 100$ (b) $T = 500$ (c) $T = 1,000$

図 2: Required amount of memory

図 2 に必要メモリ量を T の 3 つの各場合に対して示す。 T のいずれの場合も、hash, bst, DAG の順で必要メモリ量が多い。これはハッシュチェーン法ではハッシュテーブルで最大の容量をもつバケットの容量が全てのバケットで確保されるため大量のメモリを必要とするが、DAG ではほかのテーブルで管理する必要がないため最も必要メモリ量が少ない。また、 T が増加しても DAG と bst 間のチップ選択による必要メモリ量に差はあまり見られない。しかしハッシュチェーン法の $T = 100$ では URS が最も必要メモリ量が大きく、 $T = 500$ では URW が大きい。 $T = 1,000$ では WRW が 3 つのチップ選択アルゴリズムの中で最も必要メモリ量が多い。

5. まとめ 本稿では、IoT の seller による騒音データ (コンテンツ) のアップロードに際し、登録データの改ざんが困難な分散台帳技術の一つである IOTA で騒音データを管理し、micropayment の処理を IOTA 上で行うことで取引手数料を要しない決済方式を提案した。また IOTA 内でコンテンツ名を検索する四つの検索手法の検索時間とチップ選択アルゴリズムごとの必要メモリ量のシミュレーション評価、比較を行った。今後は実機に評価を行う予定である。

謝辞 本研究成果は JSPS 科研費 21H03436 と 23H03388 の助成を受けたものである。ここに記して謝意を表す。

参考文献

[1] S. Popov, et al., Equilibria in the Tangle, Computers & Industrial Engineering, 136, pp.160-172, Oct. 2019.