Computer Security Symposium 2019
21 - 24 October 2019

ビットコインにおける 手数料を考慮したオフチェーントランザクションの管理

長嶺 隆寛[1,a)]　松浦 幹太[1,b)]

概要：ビットコインは単位時間当たりに処理できるトランザクション数が少ないというスケーラビリティ問題を抱えている．スケーラビリティ問題改善に向けて，一部のトランザクションをブロックチェーンの外で管理することでブロックチェーン上に記載するトランザクション数を減らすペイメントチャネルという手法が提案されている．先行研究にて，トランザクションに時間制約を設けるタイムロックという機能を利用してトラザクション置換を行うペイメントチャネルが提案されているが，これはトランザクションの優先度を決定付ける手数料を考慮していない．
本稿では，適切な手数料を設定しないとタイムロックを利用したオフチェーントランザクションの管理が期待通りに動作せず，チャネル利用者の資金が奪われる可能性があることを指摘する．また，これを解決するために必要な手数料を追加する手法について述べ，チャネル利用者が支払う手数料の公平性について検討する．

キーワード：ブロックチェーン，ビットコイン，ペイメントチャネル

Managing Off-Chain Transactions Considering Fees in Bitcoin

Takahiro Nagamine[1,a)]　Kanta Matsuura[1,b)]

Abstract: Bitcoin suffers from a scalability problem that it cannot process lots of transactions per unit time. To improve the scalability problem, payment channels, where transactions are processed outside of the blockchain, have been proposed to reduce the number of on-chain transactions. Payment channels that replace transactions by utilizing timelocks, which set time constraints on transactions, have been proposed in previous researches, however, they have not considered transaction fees which determine the transaction priority.
In this paper, we point out that the management of off-chain transactions utilizing timelocks doesn't work appropriately without appropriate fees. Besides, we describe the method of adding fees to solve the above-mentioned problem and discuss the fairness of fees paid by channel users.

Keywords: Blockchain, Bitcoin, Payment Channel

1. はじめに

ビットコイン [19] はブロックチェーンを基盤システムとして設計されているが，ブロックチェーンのトランザクション処理速度はブロックサイズとブロックの生成間隔によって制限される．ビットコインでは，ブロックサイズの上限が 1MB[*1]，ブロックの生成間隔が約 10 分なので，1 秒間あたり最大で 10 トランザクション程度しか処理することができない．このように，ビットコインは単位時間あたりに処理できるトランザクション数が少ないというスケーラビリティ問題を抱えている．[7]

[1] 東京大学生産技術研究所
Institute of Industrial Science, The University of Tokyo
[a)] nagamine@iis.u-tokyo.ac.jp
[b)] kanta@iis.u-tokyo.ac.jp

[*1] 厳密には 1MB のブロックサイズで制限されず，SegWit[18] にて導入されたブロックウェイトで制限される．

スケーラビリティ問題解決に向けて，長きに渡りブロックサイズを大きくするという議論がなされている [5][13][21] が，これらの提案は問題を伴う．例えば，ブロックサイズを大きくするとノードに高い処理能力が要求されるため，ビットコインネットワークが特定のノードに集中してしまう可能性があることや，ブロック伝播遅延によって引き起こるチェーンの分岐の増加が問題点として挙げられる．[9] また，ブロック生成間隔の短縮においても同様にチェーン分岐の増加によってシステムが不安定になる．[11] このように，ブロックチェーンのプロトコル変更は様々な問題が発生するため，これらのプロトコル変更を必要としないブロックチェーンの外 (オフチェーン) のアプローチが数多く提案されている．[20][17][16] これらのアプローチでは，トランザクションの処理をブロックチェーンの外で行い，複数の処理を集約したトランザクションのみをブロックチェーンに記載する．このようにすることで，ブロックチェーンに記載されるトランザクションの数が減少し，スケーラビリティ問題改善に繋がる．本稿では，2 者がブロックチェーンの外で資金交換を行うペイメントチャネルに着目し，特にタイムロックを使用してトランザクション置換を行うペイメントチャネル [10][6] を取り上げる．ペイメントチャネルではトランザクションの管理をブロックチェーンの外で行うので，チャネル内で安全なトランザクションの管理を行わなければならない．具体的には，過去の資金残高を反映したトランザクションを，最新の資金残高を反映したトランザクションで置換する必要がある．

タイムロックを使用してトランザクション置換を行うペイメントチャネルは，トランザクション作成時と有効時に時間差があるという特徴を持つ．この時間内に手数料相場が高騰した場合，作成されたトランザクションは相対的に低い手数料を保持していることになり，トランザクションは優先的にブロックに追加されない．本稿では，このような場合タイムロックを使用したトランザクション置換が期待通りに動作せず，チャネルを構築しているユーザーの資金が盗まれる可能性があることを指摘する．

本稿の構成は以下の通りである．2 章でビットコインの仕組みについて，3 章でペイメントチャネルの仕組みについて説明し，4 章にてペイメントチャネルの脆弱性を指摘する．5 章でトランザクションの優先度を高める解決策を提案し，6 章でユーザーが支払う手数料の公平性について議論する．最後に，7 章にて結論を述べる．

2. ビットコイン

ビットコイン [19] は中央集権的な管理者を排除した暗号通貨である．ビットコインシステムは P2P ネットワーク上に構築されており，トランザクションはブロックチェーンによって管理される．本章では，トランザクションの構成とそれに類する機能を紹介する．

2.1 トランザクション

トランザクションとは，ビットコインの送金情報をビットコインネットワークに示すものである．トランザクションは主にインプットとアウトプットから構成され，インプットには送金元情報, アウトプットには送金先情報がビットコインアドレスとして格納される．図 1 に，5BTC 保持しているアリスが, ボブに 2BTC 送金するトランザクションを示す．全ての資金をインプットからアウトプットに移動させるので，アリスのお釣りはアウトプットに記載される．図 1 では省略しているが，インプットとアウトプットの差額がマイナーが受け取る手数料となる．

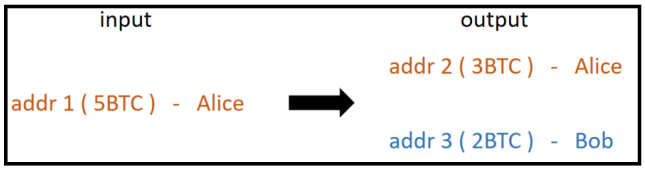

図 1 ビットコイントランザクション

未使用のアウトプット (UTXO: unspent transaction output) は，将来のトランザクションのインプットとなり得るのでビットコインの資金そのものである．アウトプットにはその所有者の公開鍵の情報が格納される．そして，そのUTXO を使用する際に所有者は秘密鍵でトランザクションに対し署名を行う．このようにすることで，UTXO は所有者以外に使用されることがなく，作成したトランザクションは他者に改ざんされない．

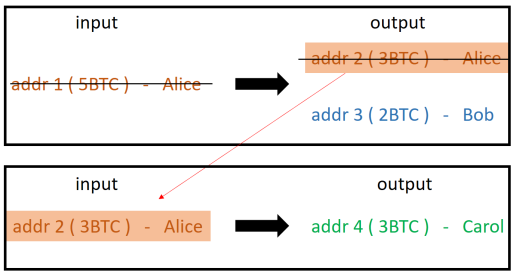

図 2 UTXO が使用される様子

2.2 タイムロック

タイムロックとは，指定された時間に達するまでビットコインの送金を制限する機能である．トランザクションが有効になる時間に関して，絶対時間で制御する locktime や，インプットが参照するアウトプットが承認を終えてからの相対時間で制御する sequence number[12] などがある．タイムロックが設定されたトランザクションは，その指定された時間[*2]に達するまでブロックに追加されず，またビットコインネットワークをリレーしない．

[*2] median time past[14] と比較される

2.3 マルチシグネチャ

マルチシグネチャ [4] とは，1 つのトランザクションの署名に対し複数の秘密鍵を割り当てる手法のことである．m 個の秘密鍵の内，n 個の秘密鍵を使用することで UTXO をアンロックするようなマルチシグネチャのことを m-of-n マルチシグと呼ぶ．ペイメントチャネルでは，チャネルを構築している 2 者の合意のもとでトランザクションが作成されるので，2-of-2 マルチシグが使用される．

3. ペイメントチャネル

3.1 ペイメントチャネルの概要

ペイメントチャネルとは，2 者がブロックチェーンの外で複数のトランザクションを管理する (資金の交換を行う) 手法である．Trusted execution environment を使用したペイメントチャネル [17] や，ペナルティを設定することでユーザーに不正を行う動機を与えないペイメントチャネル [20]，後述するタイムロックを使用してオフチェーントランザクションを管理するペイメントチャネル [10][6] などが提案されている．ペイメントチャネルでは，複数のトランザクションをブロックチェーンの外で管理し，それらの結果を集約したトランザクションのみをブロックチェーンに記載する．このようにすることで，ブロックチェーンに記載されるトランザクション数が減少し，ユーザーは手数料削減や送金時間短縮，ビットコインはスケーラビリティ問題改善に繋がる．ペイメントチャネル構築から終了までのプロセスを以下に示す．(2) の実装はペイメントチャネルにより異なるが，ここでは基本的なモデルを紹介する．

(1) チャネルの構築：2-of-2 マルチシグにチャネル内で使用する資金をデポジットするファンディングトランザクションを作成し，ビットコインネットワークにブロードキャストする．[*3]

(2) チャネル内での資金交換：2-of-2 マルチシグをインプットにとり，各ユーザーをアウトプットにとるコミットメントトランザクションを作成する．アウトプットの額が各ユーザーの残高となる．

(3) チャネルの終了：最終的な残高を反映したセトルメントトランザクションを作成し，ビットコインネットワークにブロードキャストする．

資金の交換は，コミットメントトランザクションを発行することで行われる．各ユーザーのアウトプットの額を更新したコミットメントトランザクションを発行することで，更新された額だけ資金の交換が行われることになる．

*3 片方のユーザーが音信不通になった場合，デポジットした資金が永久にロックされるので，ファンディングトランザクションをブロードキャストする前に，最初のコミットメントトランザクションを作成する．

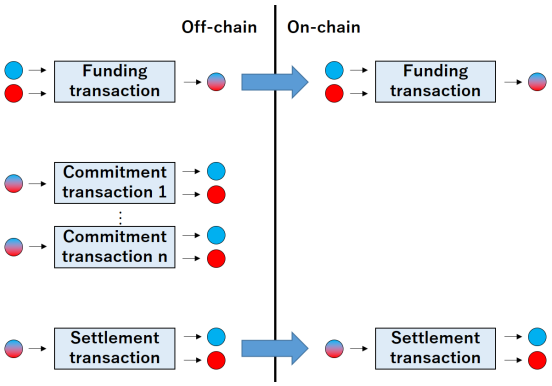

図 3 ペイメントチャネルの概要

また，コミットメントトランザクションは 2-of-2 マルチシグをインプットにとるので，資金の交換は必ず両者の合意のもと (署名のもと) 行われる．複数回トランザクションを発行しても，ブロックチェーン上にはファンディングトランザクションとセトルメントトランザクションの 2 つしか記載されない．片方のユーザーが合意せずセトルメントトランザクションを作成できない場合は，最新のコミットメントトランザクションをブロードキャストすることでチャネルを終了することができる．

チャネルを使用する上で注意するべきことは，オフチェーンで管理されるコミットメントトランザクションの取り扱いである．具体的には，**最新のコミットメントトランザクションのみを有効にする仕組み**が必要となる．過去のコミットメントトランザクションには過去の残高が反映されているので，これが有効になると最新の取引が反映されず，最新のコミットメントトランザクションで資金を受け取るユーザーは相手に資金を盗まれる事態が発生する．例として，アリスとボブがペイメントチャネルで資金交換を行い，両者の資金残高が図 4 のように変化した状況を考える．

	Balance(BTC)		description
	Alice	Bob	
initial	0.5	0.5	-
commitment transaction 1	0.1	0.9	Alice sends 0.4BTC to Bob
commitment transaction 2	0.9	0.1	Bob sends 0.8BTC to Alice

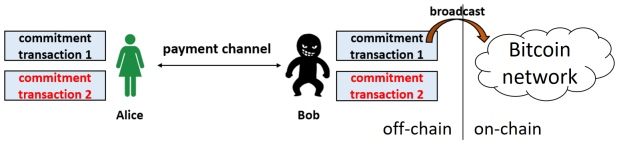

図 4 ペイメントチャネルで資金が盗まれる様子

コミットメントトランザクション 2 に現在の残高が反映されているので，これが有効になることが期待される．この時の両者の残高はアリスが 0.9BTC, ボブが 0.1BTC である．悪意を持ったボブがコミットメントトランザクション 1 をビットコインネットワークにブロードキャストし，これがブロックに追加されると，ブロックチェーン上には

アリスの残高が 0.1BTC, ボブの残高が 0.9BTC という情報が記載される．この時，アリスはボブに 0.8BTC 奪われる事態が発生するので，先述の通りペイメントチャネルでは最新のコミットメントトランザクションのみを有効にする仕組みが必要となる．

3.2 タイムロックを使用してトランザクション置換を行うペイメントチャネル

3.1 章にて，ペイメントチャネルでは過去のコミットメントトランザクションを無効にし，最新のコミットメントトランザクションのみを有効にする仕組みが必要であることを述べた．トランザクションを無効にする仕組みとして，タイムロックがある．

3.2.1 タイムロックを使用したトランザクションの置換

過去のトランザクションを無効にするには，最新のトランザクションのタイムロックに，過去のトランザクションのタイムロックに設定した値よりも早い値を設定すれば良い．

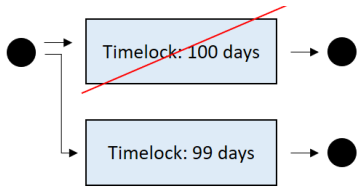

図 5　タイムロックを使用したトランザクションの置換

図 5 に，タイムロックを使用してトランザクションを無効にする様子を示す．図 5 では，100 日のタイムロックが設定されたトランザクションを，99 日のタイムロックが設定されたトランザクションで無効にしている．これは，仮に上のトランザクションがブロードキャストされたとしても，これが有効になるのは 100 日後なので，これよりも早く，99 日後に有効になる下のトランザクションの方が優先的にブロックに追加されるからである．

3.2.2 ペイメントチャネルへの応用

タイムロックを使用したトランザクションの置換を使用して，オフチェーントランザクションの管理を行うペイメントチャネルが提案されている．[10][6]

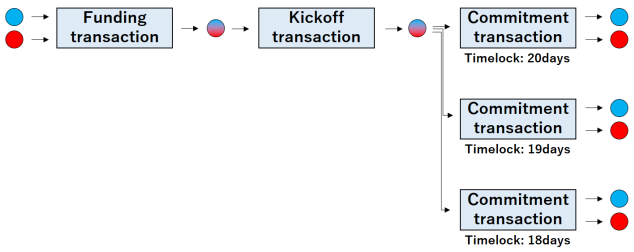

図 6　相対タイムロックを使用したコミットメントトランザクションの置換

図 6 のペイメントチャネルでは，コミットメントトランザクションの置換をタイムロックによって行っている．相対タイムロックを使用する場合は，相対値の起点となるキックオフトランザクションも作成する．キックオフトランザクションが承認されてからの相対時間によって，タイムロックは計算される．チャネル内で使用するタイムロックの最大値を T_{max}，最小値を T_{min}，間隔を ΔT とすると，コミットメントトランザクションの更新回数は最大で $(T_{max} - T_{min})/\Delta T$ となる．一般的に，二重支払い防止のためにトランザクションは 6 承認以上を必要とするため，ΔT の最小値は 1 時間とされている．[10] 図 7 のように，タイムロックで無効化するコミットメントトランザクションの層を追加することで，コミットメントトランザクションの更新回数を増やすことができる．このような多層のコミットメントトランザクションの構造は invalidation tree と呼ばれる．[10] 図 7 では，タイムロックが 1 番早いトランザクションの枝 (Timelock:18days - Timelock:18days) のみが有効となる．

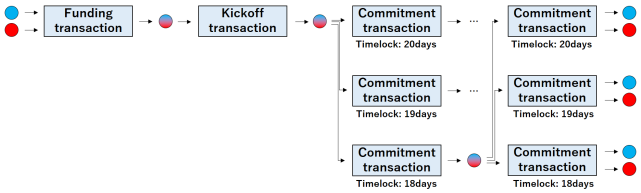

図 7　相対タイムロックを使用した invalidation tree

チャネルの終了は，2 者が協力する場合と，協力しない場合で異なる．協力する場合は最新の残高を反映したセトルメントトランザクションを作成し，これをビットコインネットワークにブロードキャストする．協力しない場合は，キックオフトランザクションと最新のコミットメントトランザクションをブロードキャストする．この時，コミットメントトランザクションにはタイムロックが設定されているため，タイムロックが有効になるまで 2 者は資金を使用することができない．

4. 問題提起

3.2 章で紹介したペイメントチャネル [10][6] は非協力的なチャネル終了の場合，コミットメントトランザクションを作成してからそれが有効になるまで時間差があるという特徴を持つ．また，各トランザクションに設定する手数料を考慮していない．本章では，トランザクションの優先度と手数料を確認し，[10][6] で提案されているオフチェーントランザクションの管理は正常に動作しない場合があることを指摘する．

4.1 トランザクションの優先度

ビットコインの古いバージョンでは，Coin Age Prior-

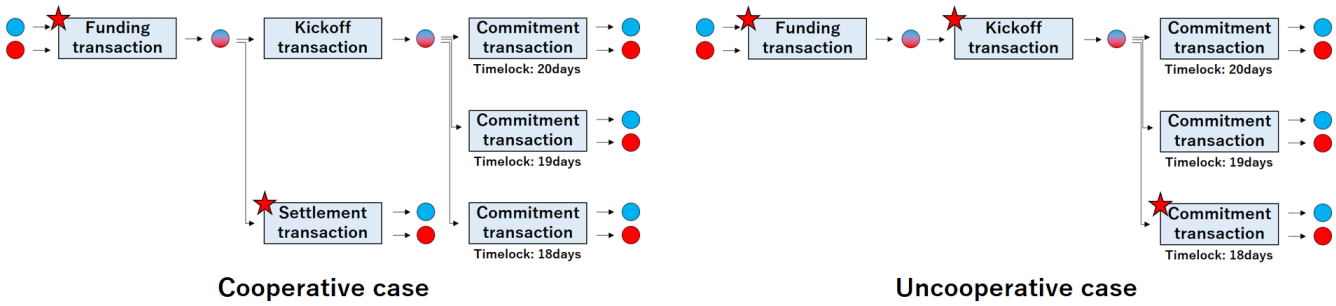

図 8 協力的，非協力的なチャネル終了の様子．星マークのトランザクションがブロードキャストされる．

ity[*4]という仕組みでインプットの額と年齢が高いトランザクションが優先的にブロックに追加されていたが，Bitcoin Core 0.15.0 より Coin Age Priority は廃止された．[*5] マイナーはブロックに追加するトランザクションを自由に選択することができるので，最大限の利益を得るために 1 バイト当たりの手数料が高いトランザクションを優先的にブロックに追加する．[15]

4.2 トランザクション手数料

ビットコインのトランザクション手数料の推移を図 9 に示す．図 9 において，各色はトランザクションが n ブロック以内にマイニングされるために必要な手数料を表している．手数料が高いトランザクションほど優先度が高いので，1 ブロック以内にマイニングされるために必要な手数料 (青) は，6 ブロック以内にマイニングされるために必要な手数料 (橙) よりも高いことが読み取れる．

注目するべきは，そのボラティリティの高さである．2017 年 12 月をピークに，数か月で平均手数料が 100 倍近く下がっている．トランザクションは手数料に応じて優先度が決定付けられるので，ビットコインネットワークの混雑度に応じて手数料が変化する．

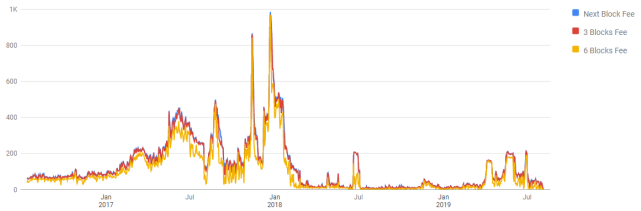

図 9 トランザクションの平均手数料．x 軸は日付，y 軸はトランザクション手数料 (satoshis/byte) を表している．[1]

4.3 ペイメントチャネルの脆弱性

[10][6] は非協力的なチャネル終了の場合，コミットメントトランザクション作成時に設定された手数料が有効時に最適であるとは限らない．コミットメントトランザクション作成後に手数料相場が高騰した場合，相対的に低い手数料を持つコミットメントトランザクションは優先的にブロックに追加されない場合がある．

最新のコミットメントトランザクション (以下，$ComTX_n$) がブロックに追加されず ΔT 経過した場合，2 番目に新しいコミットメントトランザクション (以下，$ComTX_{n-1}$) が有効になるので，同一 UTXO を参照する $ComTX_n$ と $ComTX_{n-1}$ との間で競合が発生する．$ComTX_{n-1}$ がブロックに追加されると，最新の取引はブロックチェーンに反映されないので，最新の取引で資金を受け取るユーザーは取引相手に資金を盗まれるという事態が発生する．

一般的に，$ComTX_{n-1}$ に $ComTX_n$ よりも高い手数料が設定されていない限り，メモリプールにおいて $ComTX_n$ は $ComTX_{n-1}$ に置換されない．[8] しかし，$ComTX_{n-1}$ は確かにマイニング可能な状態であり，最新の取引で資金を送金するユーザーが意図的に $ComTX_{n-1}$ をマイニングすることも考えられるので，一定のリスクは残る．

5. 解決策

4.3 章で指摘した問題点は，コミットメントトランザクションに高額な手数料を設定することや，様々な手数料を設定したコミットメントトランザクションを複数作成することで解決できる．しかし，前者は非協力的なチャネル終了の際に必要以上に高い手数料を払わなければいけないリスクがあり，後者はより多くのトランザクションを作成しなければいけないデメリットがある．また，将来の手数料相場が予測できない以上，設定する手数料の上限値を決定することが難しい．ΔT を大きくすることも解決策として考えられるが，ΔT の時間内に優先度の低いトランザクションがブロックに追加されるとは限らないし，ΔT を大きくすることで作成できるコミットメントトランザクションの数が少なくなってしまう．

本章では，解決策としてコミットメントトランザクションに手数料を追加する手法を考察する．通常は，手数料を変更してトランザクションを作り直すことで，手数料を追

*4 コンセンサスルールではなく，マイナーの方針選択
*5 https://github.com/bitcoin/bitcoin/pull/9602

加することができる．しかし，2-of-2 マルチシグを参照するコミットメントトランザクションは作成に 2 者の署名を必要とするため，非協力的なチャネル終了の際にユーザーは単独でコミットメントトランザクションを作り直すことができない．ここでは，ユーザーが単独でもコミットメントトランザクションに手数料を追加できる手法について考察する．

5.1 SIGHASH_ANYONECANPAY

SIGHASH[2] とはトランザクションの署名範囲を示すものである．デフォルトは SIGHASH_ALL であり，全てのインプットとアウトプットが署名される．署名対象のデータを変更することは不可能なので，SIGHASH_ALL で署名されたトランザクションに手数料を追加することはできない．

これに対し SIGHASH_ANYONECANPAY は，追加のインプットを可能にする SIGHASH である．SIGHASH_ALL|SIGHASH_ANYONECANPAY のように他の SIGHASH と組み合わせて使用される．SIGHASH_ALL|SIGHASH_ANYONECANPAY は 1 つのインプットと全てのアウトプットを署名対象とし，第三者は他のインプットを追加したり削除したりすることができる．コミットメントトランザクションを SIGHASH_ANYONECANPAY で署名することで，非協力的なチャネル終了の場合でもユーザーは単独で手数料を追加することができる．

5.2 Child Pays for Parent

Child Pays for Parent(CPFP)[3] とは，トランザクションの優先度を高めるために，そのトランザクション (親) のアウトプットを参照する子トランザクションを作成し，子トランザクションに高額な手数料を設定する手法である．高い報酬を得るために子トランザクションを承認したいマイナーは，これを実現するために親トランザクションも承認する必要があるので，結果的に親トランザクションの優先度は高くなる．

ペイメントチャネルでは，コミットメントトランザクションのアウトプットを参照するトランザクションを作成しこれに高額な手数料を設定することで，コミットメントトランザクションの優先度を高めることができる．

6. 手数料の公平性の問題

5 章にて，トランザクションの優先度を高めるために手数料を追加する手法を述べたが，手数料を追加する人はその手数料分，資金を余分に支払わなければいけないという問題が発生する．手数料を追加する人とは，$ComTX_{n-1}$ がブロックに追加されることを阻止したい人，即ち $ComTX_n$ で資金を受け取るユーザー（受取人）である．ここでは，コミットメントトランザクションで交換する資金は追加手数料よりも高く，受取人は $ComTX_n$ をブロックに追加する動機が十分にあると仮定する．追加手数料がコミットメントトランザクションで交換する資金より高い場合は，2 者ともに手数料を追加する動機がなく複数のコミットメントトランザクションが有効になる可能性があるので，チャネル内で安全な資金交換を行うことができない．

非協力的なチャネル終了の場合，$ComTX_n$ で資金を送るユーザー（送金人）は追加手数料を支払う必要がない．$ComTX_n$ 作成後に手数料が高騰し，セトルメントトランザクションを作成する際に高額な手数料が必要な場合，送金人は非協力的なチャネル終了を選択する方が総手数料を安く抑えられることがある．例として，ファンディングトランザクション，キックオフトランザクション，$ComTX_n$ 作成時の手数料相場が xBTC，その後手数料相場が高騰し 10xBTC になった状況を考える．各トランザクションに設定する手数料はユーザー間で折半し，トランザクション作成時の手数料相場を適用するものとする．この時点でチャネルを終了する場合，各ユーザーが支払う総手数料は表 1 のようになる．

表 1 各ユーザーが支払う総手数料

	送金人	受取人
協力時	(x + 10x)/2	(x + 10x)/2
非協力時	(x + x + x)/2	(x + x + x)/2 + 追加手数料

$ComTX_n$ 有効時に手数料相場が下がれば問題ないが，10xBTC のままだと追加手数料は 9xBTC となる．この時，送金人は手数料を節約できる一方で，受取人は多くの手数料負担を強いられることになり，手数料の公平性の問題が生じる．

6.1 コミットメントトランザクションの手数料

公平な手数料負担を実現するためには，ユーザーに非協力的なチャネル終了を選択する動機を与えなければよい．具体的には，協力的なチャネル終了に必要な手数料が，非協力的なチャネル終了に必要な手数料を下回ればよい．コミットメントトランザクションに設定した手数料が，時間と共に変化する手数料相場と乖離することで問題が生じるので，コミットメントトランザクションの作成間隔が大きな鍵となる．なお，コミットメントトランザクションに設定する手数料は，作成時の手数料相場を適用することを考える．

コミットメントトランザクションの作成間隔が大きい場合，基本的に対応は難しい．これは，作成間隔が小さい場合に比べ，コミットメントトランザクションに設定した手数料が手数料相場と大きく乖離する可能性が高いからである．コミットメントトランザクションに高い手数料を設定

することで，非協力的なチャネル終了を選択する動機を与えないことはできるが，5 章で言及した通り，どれほど高い手数料を設定するべきなのか決定することが難しく，また，非協力的なチャネル終了の際に必要以上に高い手数料を払わなければいけないリスクがある．コミットメントトランザクションの作成間隔が小さい場合は，送金人が受取人よりも多くの手数料を負担することで協力的なチャネル終了を選択する動機を与えることができる．手数料の負担額が多いことから，送金人が非協力的なチャネル終了を選択する動機はないし，受取人も追加手数料を払うリスクがあることから非協力的なチャネル終了を選択する動機はない．

7. おわりに

本稿では，手数料変動を考慮すると，タイムロックを使用してトランザクション置換を行うペイメントチャネルは正常に動作しない可能性があることを指摘した．タイムロックを使用してトランザクション置換を行うペイメントチャネルは，コミットメントトランザクション作成時と有効時に大きな時間差がある．非協力的なチャネル終了の場合，手数料相場が高騰すると相対的に低い手数料を持つコミットメントトランザクションは優先的にブロックに追加されず，複数のコミットメントトランザクションが有効になる可能性がある．従って，このオフチェーントランザクションの管理は不十分である．

解決策として，SIGHASH_ANYONECANPAY や CPFP を使用してトランザクションの優先度を高める手法を考察し，ユーザーが負担する手数料の公正性について述べた．本稿で考察した解決策は，過去のコミットメントトランザクションが有効になることを阻止する手法であり，依然としてコミットメントトランザクションの置換は十分に機能していない．公平かつ安全な資金交換を達成するには，根本的なペイメントチャネルの設計の見直しが必要であると考える．

謝辞 本研究の一部は，JSPS 科研費 17KT0081 の助成を受けたものである．ここに感謝します．

参考文献

[1] `https://bitcoinfees.info/`.
[2] `https://bitcoin.org/en/transactions-guide#signature-hash-types`.
[3] `https://bitcoin.org/en/glossary/cpfp`.
[4] Gavin Andresen. BIP11: M-of-N Standard Transactions, 2011. `https://github.com/bitcoin/bips/blob/master/bip-0011.mediawiki`.
[5] Gavin Andresen. BIP101: Increase maximum block size, 2015. `https://github.com/bitcoin/bips/blob/master/bip-0101.mediawiki`.
[6] Conrad Burchert, Christian Decker, and Roger Wattenhofer. Scalable funding of Bitcoin micropayment channel networks. *Royal Society open science*, 5(8):180089, 2018.
[7] Kyle Croman, Christian Decker, Ittay Eyal, Adem Efe Gencer, Ari Juels, Ahmed Kosba, Andrew Miller, Prateek Saxena, Elaine Shi, Emin Gün Sirer, et al. On Scaling Decentralized Blockchains. In *International Conference on Financial Cryptography and Data Security*, pages 106–125. Springer, 2016.
[8] David A. Harding and Peter Todd. BIP125: Opt-in Full Replace-by-Fee Signaling, 2015. `https://github.com/bitcoin/bips/blob/master/bip-0125.mediawiki`.
[9] Christian Decker and Roger Wattenhofer. Information Propagation in the Bitcoin Network. In *IEEE P2P 2013 Proceedings*, pages 1–10. IEEE, 2013.
[10] Christian Decker and Roger Wattenhofer. A Fast and Scalable Payment Network with Bitcoin Duplex Micropayment Channels. In *Symposium on Self-Stabilizing Systems*, pages 3–18. Springer, 2015.
[11] Ittay Eyal, Adem Efe Gencer, Emin Gün Sirer, and Robbert Van Renesse. Bitcoin-NG: A Scalable Blockchain Protocol. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, pages 45–59, 2016.
[12] Mark Friedenbach, BtcDrak, Nicolas Dorier, and kinoshitajona. BIP68: Relative lock-time using consensus-enforced sequence number, 2015. `https://github.com/bitcoin/bips/blob/master/bip-0068.mediawiki`.
[13] Jeff Garzik. BIP102: Block size increase to 2MB, 2015. `https://github.com/bitcoin/bips/blob/master/bip-0102.mediawiki`.
[14] Thomas Kerin and Mark Friedenbach. BIP113: Median time-past as endpoint for lock-time calculations, 2015. `https://github.com/bitcoin/bips/blob/master/bip-0113.mediawiki`.
[15] David Koops. Predicting the confirmation time of bitcoin transactions. *arXiv preprint arXiv:1809.10596*, 2018.
[16] Joshua Lind, Ittay Eyal, Florian Kelbert, Oded Naor, Peter Pietzuch, and Emin Gun Sirer. Teechain: Scalable blockchain payments using trusted execution environments. *arXiv preprint arXiv:1707.05454*, 2017.
[17] Joshua Lind, Ittay Eyal, Peter Pietzuch, and Emin Gün Sirer. Teechan: Payment Channels Using Trusted Execution Environments. *arXiv preprint arXiv:1612.07766*, 2016.
[18] Eric Lombrozo, Johnson Lau, and Pieter Wuille. BIP141: Segregated Witness, 2015. `https://github.com/bitcoin/bips/blob/master/bip-0141.mediawiki`.
[19] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008.
[20] Joseph Poon and Thaddeus Dryja. The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments. *https://lightning.network/lightning-network-paper.pdf*, 2016.
[21] Pieter Wuille. BIP103: Block size following technological growth, 2015. `https://github.com/bitcoin/bips/blob/master/bip-0103.mediawiki`.