

P2P コンテンツ配信ネットワークにおけるランダム性の排除による通信性能向上に関する研究

研究代表者 大坐 晶 智 電気通信大学大学院情報システム学研究科 准教授

1 研究の目的

大きなファイル、動画配信する方式として、BitTorrent が幅広く用いられ、インターネットトラフィックの半数近くを占めるという報告もある。その人気の理由として、高速なコンテンツ配信方式があげられるが、ファイルをダウンロードする際の戦略に改善の余地がある。BitTorrent では、ファイルをダウンロードする際に、ファイルをピースに分割し、ピースをダウンロードするピア (PC) に配布する。配布されるピースはダウンロードするピア毎に異なり、ファイルを完成させるために必要なピースをピア間で交換しながら、ファイルを完成させる。ファイルの配布元の負荷の集中をさせることにより、それぞれの PC でのファイルのダウンロード時間を短縮させている。ファイルを完成させるまでに、ピースを集める戦略と、ピアを選択する戦略があるが、このどちらにもランダムにピース、ピアを選ぶ戦略が入っていることにより、同一のピースが何度も同一経路をとったり、遠いピアと通信しようとしたりする。これは、アプリケーション層レベルで構成させる BitTorrent は、IP ネットワークの構成を把握せずに通信をおこなうことと (図 1)、ピース収集戦略とピア選択戦略の間で協調動作をせず、バラバラに動作するためである。

これまでに、これらの P2P における問題がネットワークへ多大な負荷をかけてきた。これらの問題を解決するためには、バックボーンネットワークの帯域を圧迫する冗長なトラフィックを削減する制御が必要である。文献[1]では P2P によるバックボーンネットワークの圧迫によるコストの削減方式を提案している。クロス ISP 間のトラフィックを削減するために ISP 内にユーザ間の距離を測るためのサーバを用意し、互いに距離の近いユーザ同士を接続する方式である。この方式により ISP 間トラフィックの改善はされた。しかしながら、ローカライゼーションにより、ピースを取得する範囲が制限され、ダウンロード速度の低下が起きるという問題がある[2]。

そこで、本研究調査では、ピース収集戦略とピア選択戦略の関係をわかりやすくするピース収集方式を明らかにし、ピア間の協調動作が促進するような戦略を明らかにすることで、さらなる高速かつ高効率のコンテンツ配信方式を開発する。

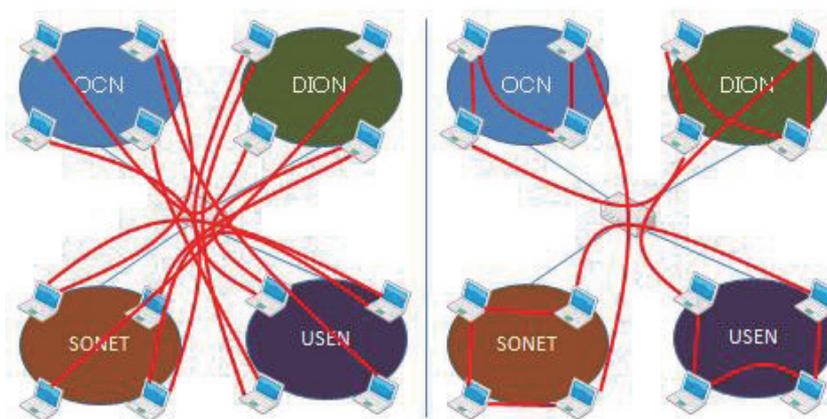


図 1 ランダムなピア選択による冗長トラフィックの発生

2 BitTorrent における塊を考慮したダウンロード方式

2-1 提案方式概要

提案方式では、図 2 のようにピースをいくつか束ねた塊にして扱う。塊の状態でのファイルのダウンロード

を行った場合、既存方式の問題である、誰がどのピアへアップロードし、どのピアからダウンロードすれば良いのかがわかりやすくなったため、ピア選択戦略での塊を利用した戦略が立てられる。ピース選択では塊の完成を最優先にピースのダウンロードを行う。後述するピア選択戦略が互いの塊の持ち合いを見て相手ピアを選択する方式を取っており、ピア選択する際に必要な「塊」をできるだけ早く入手する必要がある。これを実現するために本方式ではトラックが各々のASが所持している塊を管理することとした。ピアはトラックの管理記録からどのASにどの塊があるか、どの塊が不足しているのかが確認でき、ピース選択の際に自分がどの塊を担当してダウンロードをすれば効率が良いか判断出来る。既存の方式にあるピース選択に用いられている Random-first 戦略を廃止し、トラックを利用した方式を用いることでピースの重複を防ぐ。

ピア選択戦略では楽天的非チョーク戦略によるランダムなピア選択を廃止し、代わりに塊を利用した戦略を取ることで、AS内のピースの重複を減らし、さらにローカライズを行うことでダウンロードにおける遅延の回避や冗長トラフィックの抑制を実現させる。初めはAS内やAS外のピアに関係なく、図2のように自分の所持していない塊を持っているピア、自分のASに無い塊を持っているピア（図2では塊1の構成するピースを持っているピア）と優先的に接続するように戦略をとり、AS内でファイルを構成する全ての塊が集まった後、図3のようにファイルのダウンロードを外のピアに頼らなくても、AS内のピアのみでダウンロードが完了できる状況ができればローカライズが働くような戦略に切り替えることでAS間トラフィックの抑制を実現する。さらにピア選択のローカライズの他に、ピース選択でもローカライズも行う。自分の所望するある一つのピースを持っているピアが複数いた場合、自分と同じASに存在するピアへ優先的に interested メッセージを送信する。他のASにいる所望のピースを所持しているピアへの interested メッセージの送信を抑えることでAS間トラフィックを抑える。

また、本方式はピア間のメッセージを追加しておらず、既存の方式のピアが参加した場合でも正常に動作する。

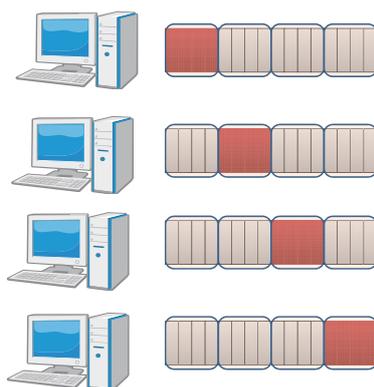


図 2 ピースを固まりで集める

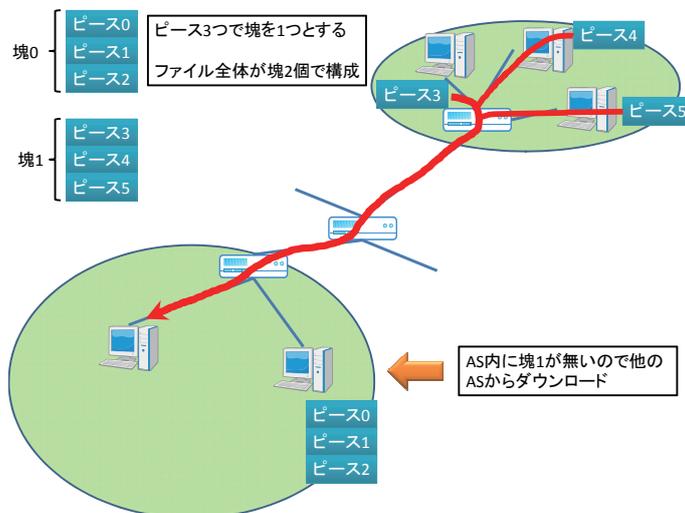


図 3 自分のASの利益になるピアと接続

2-2 固まりの管理

各々ピアが自分のASの記録だけで次にとる塊を判断すると効率性が下がる場合がある。各々のASは他のASからも塊をダウンロードできる。しかし、異なる2つのASであるAS1、AS2がそれぞれ同じ塊をシードからダウンロードした場合、互いのAS間で塊のダウンロードが行えず、シードにリクエストが集中しボトルネックとなってしまう。そこで、この問題を解決するために、トラッカーでスウォーム全体の塊の管理を行う。各々のASでダウンロードされた塊はこれまで通り自分のASの配列に記録していくが、同時にスウォーム全体の配列にも記録していく。例えばAS1のピアが0番目の塊をダウンロードしたら、自分のASの配列の0番目の要素に1を加算し、同時にスウォーム全体の配列の要素0にも1を加算し記録していく。そして、これまでピース選択の戦略としてピアは自分のASの記録を利用していたが、先にスウォーム全体の塊を確認する。スウォーム全体の塊の記録においてRarest-First戦略を取ることで、AS内においても、AS外においても重複せずにダウンロードを行うことができる。これをスウォーム全体に全ての塊が行き渡る、つまり、図4のようにスウォーム全体の塊を記録している配列に0が無くなるまで繰り返す。そして、塊がスウォーム全体に行き渡った後は通常通り、自分のASの塊の記録を確認し、不足している塊を外からダウンロードする。シードになりASに残る場合はシードが同じAS内のリーチャへ塊を優先的にアップロードする。ここまでのフローチャートを図5に示す。

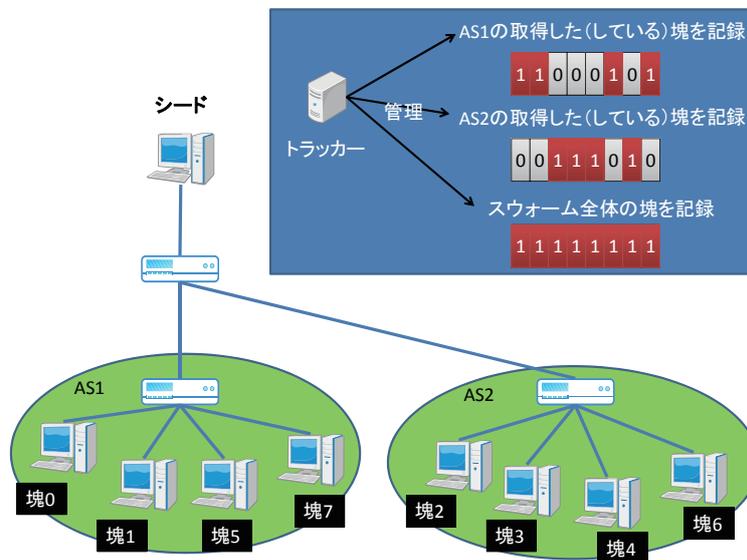


図4 スウォーム全体を見て固まりを取得

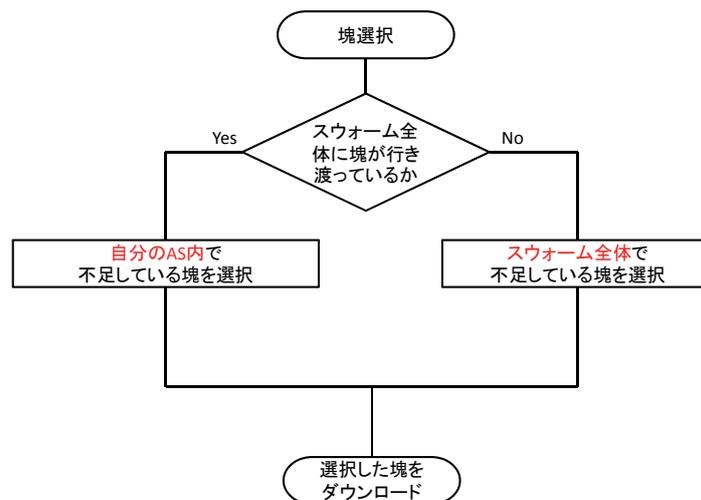


図5 塊まり選択に関するフローチャート

2-3 ピース選択戦略の改良

ピース選択戦略に関して塊を考慮した戦略に修正する。提案したピース選択戦略では Random-First を排除し、塊によるダウンロードと Rarest-First 戦略を組み合わせた方式にすることで、ピア選択戦略とリンクさせダウンロードの効率化を図る。前節では基本的なダウンロードすべき塊の優先度のつけ方は前項の塊の管理の部分で説明した。ここでは塊を構成するピースの優先度付けに関しての説明をする。

図 6 にピース選択戦略のフローチャートを示す。フローチャートの概要としては、塊を一つずつ取り出して、優先度を上げる条件に合う塊であれば、その塊を構成するピースの優先度を上げ、全ての塊を調べ終わった後、その中で最も優先度の高い塊のピースを選択する。具体的には、まず、スウォーム全体を考慮した塊の選択から始める。スウォーム全体に塊が行き渡っていない場合、AS 間で協力し合いスウォームに不足している塊を Rarest-First を用いて選択する。スウォーム全体に塊が行き渡っている場合は他の AS とシードの両方から塊をダウンロードできる状態になっているので、スウォームを考慮せず自分の AS に不足している塊を Rarest-First を用いて選択する。

塊を選択し終えた後、一度塊の中のピースへの優先度付けを行う。優先度の付け方は、まず図 6 中の(1)の先程選択した塊を構成するピースへの優先度を上げる。次に図 6 中の(2)の AS もしくはスウォーム内で自分が担当してダウンロードする塊の優先度を上げる。担当する塊は他のピアと重複することが無く、自分が最優先でダウンロードしなければならない。したがって、(1)の塊よりも優先度を上げている。これを繰り返して優先度付をピース全体について行い、その中から最も優先度の高いピースを選択し、ピース選択は終了する。

ピース選択では塊の完成を最優先にピースのダウンロードを行う。後述するピア選択戦略が互いの塊の持ち合いを見て相手ピアを選択する方式を取っており、ピア選択する際に必要な「塊」をできるだけ早く入手する必要がある。

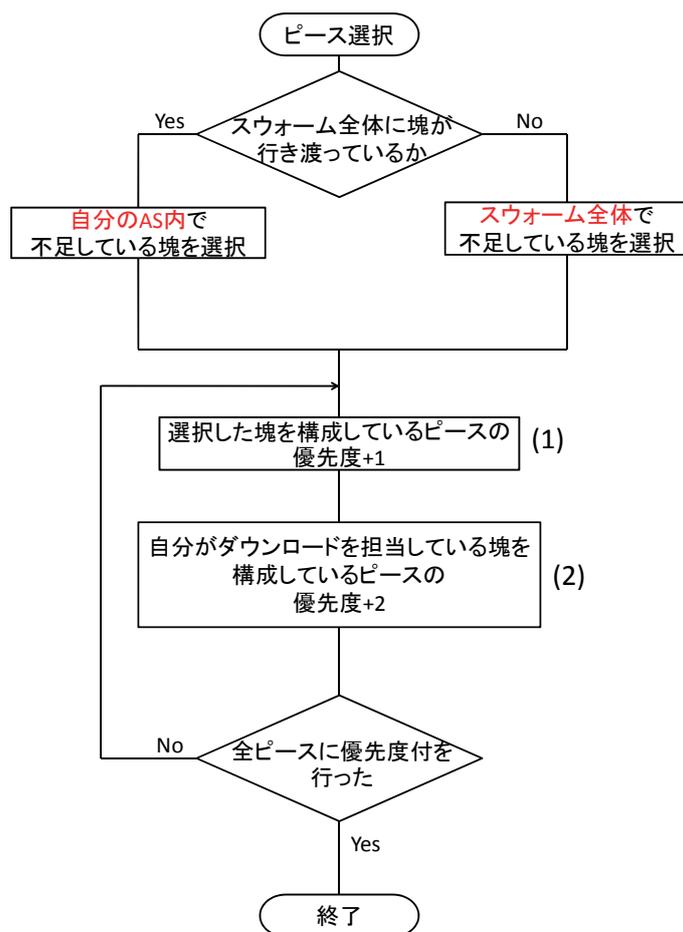


図 6 ピース選択戦略のフローチャート

2-4 ピア選択戦略の改良

ピア選択戦略でもピース選択戦略と同様に塊を考慮した戦略に修正する。既存の方式ではTit-for-tatによりピースの転送量でピアの選択をしてきた。しかし、それでは互いのピースの持ち合いによる利害関係がわかりづらかったため、ピース選択戦略で塊を作ることにより、ピア間のピースの持ち合いがわかりやすくし利害関係が明確した。この利害関係をピア選択戦略に利用する。

図 7にピア選択における非チョーク優先順位を計算するフローチャートを示す。ここでピア 1 はどのピアから塊をもらえるのか各々のピアとの利益を計算し、自分と所属する AS に利益をもたらすようなピア選択の優先度を求める。優先度が大きい方が優先的に非チョークされやすいピアとなる。自分の利益は、自ピアが所持しておらず、相手ピアが所持している塊の数の合計である。この利益の計算を、自分に対し interested メッセージを送ってきたピア全てに対し行う。自分の AS の利益は、自分の AS でまだダウンロードされていない塊の数の合計になる。AS 内で不足した塊を優先的にダウンロードすることで AS 内のピアのピース交換のみでダウンロードを完了させるためのローカライズが上手く働くように仕向ける。

また、アップロードを一切行わずダウンロードのみを行うフリーライダーの対策、そして、ローカライズがより行えるようにするための方法として優先度を 2 倍にし、非チョークの優先度を上げる方式を導入した。フリーライダーの対策には通常のピア選択で使用されている tit-for-tat 戦略が有効となる。過去に自分に対してアップロードの貢献をしているかどうかを確認することで、アップロードをしないフリーライダーはこの戦略では不利になり、貢献をしてくれたピアは有利になる。ローカライズの実現方法としては、単純に自分と同じ AS 内にいるピアであれば優先度を上げるようにした。

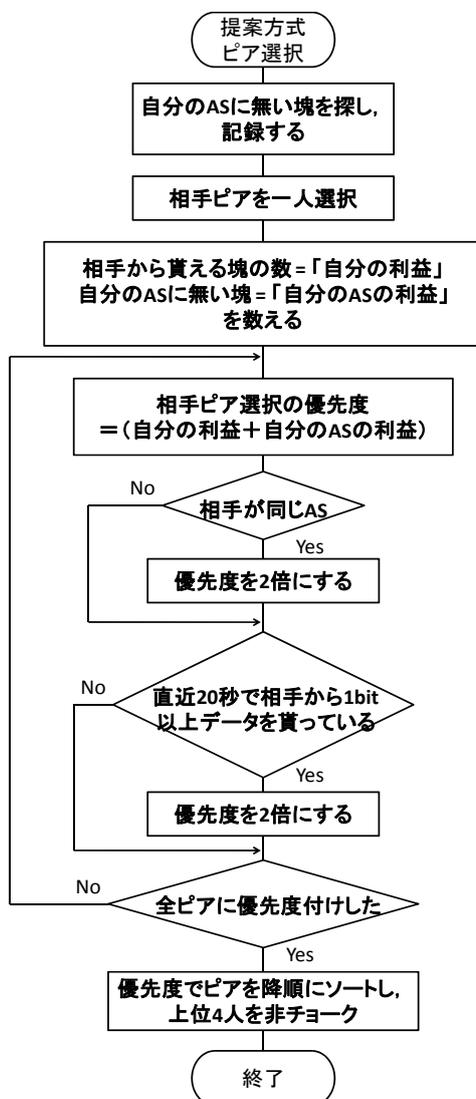


図 7 提案方式ピア選択フローチャート

2-5 ピースのローカライズ

通常、ピアは一つのピースをダウンロードし終え、ハッシュにより元のピースと同一のピースであるか確認した後に、直接接続されたピアへ have メッセージをブロードキャストし自分がそのピースを所持していることを伝える。have メッセージを受信したピアは自分がそのピースを所持していれば、そのピースは必要ないため have メッセージを送信したピアがそのピースを所持していることを自分のピアリストへ記録する。もし、そのピースを所持していなければそれをもたらすために、interested メッセージを相手ピアへ送信する。

しかし、このままでは近くにそのピースを持っているピアがいるにもかかわらず、have、interested メッセージのタイミングが前後し、自分から遠い場所にいるピアからダウンロードし、ダウンロード時間を遅くする可能性がある。

そこで、ピースにもローカライズを適用し、ダウンロード時間の減少、トラフィックの抑制を行う。方法としては、橙色のピースに関する have メッセージを受信したピア A はトラックで管理されている AS4 に関する塊の記録を確認する。その記録で橙色のピースが含まれる塊が自分の AS 内にあることが確認できた場合、have メッセージの送信者が他の AS のピアであれば interested メッセージを送信せず、自分の AS 内のピアへのみ interested メッセージを送信する。もし、自分の AS 内にそのピースが含まれる塊がなければ通常通り interested メッセージを送信し、相手から非チョークされればダウンロードを行う。図 8 にフローチャートを示す。

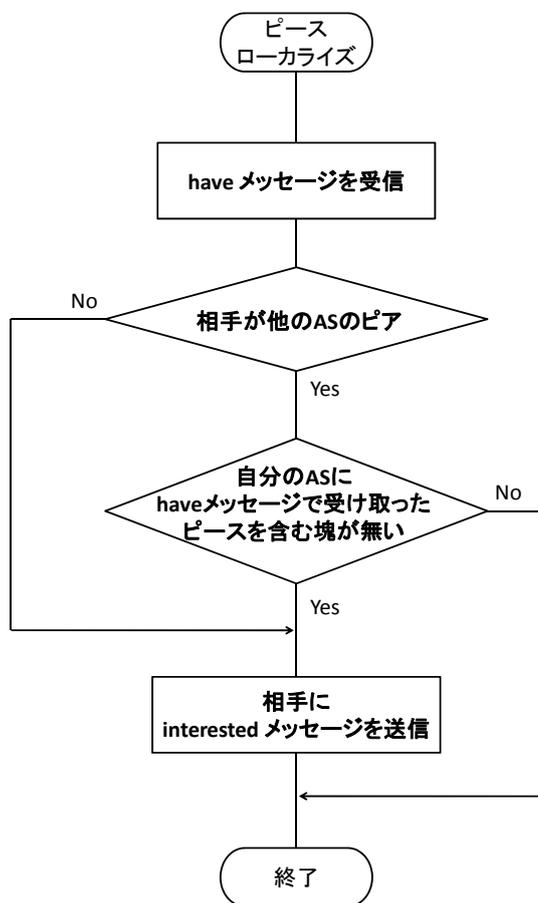


図 8 ピースのローカライズ

2-6 評価実験

実験では図 9 のトポロジを使用する。初期シードは 1 ピア、リーチャは各々の AS に同数のピア (2~10 ピア) を配置する。各 AS のピア数が 2 ピアであればスウォーム全体には初期シード 1 ピア、リーチャが 4 ピア存在することとなる。初期シードは実験開始直後に起動し、各々のリーチャは実験開始から 10 秒の間にランダムで参加する。また、各々のリーチャにはそれぞれ AS 番号が付与されており、それはトラックで管理している。ローカライズ実現の評価を行うために仮想の AS を 2 つ作成し、AS 間トラフィックを測定する。ルータ間、ルータ、ピア間の帯域、遅延を表 1 に示す。

表 1 帯域、遅延

	ルータ間	ルータ、ピア間
帯域	100Mbps	10Mbps
遅延	20ms	2ms

ルータ間の遅延はルータ、ピア間の遅延の 10 倍であるため、ローカライズを考慮せずにランダムによって他の AS のピアとの通信を行うとダウンロード時間が増加する。ダウンロードするファイルは 100MB である。実験シナリオは各々のリーチャはダウンロード完了後シードにはならず、すぐにスウォームから退出する。AS 内にシードが常にいない状況になるため、リーチャ間で上手くピースを分散させ、さらにローカライズを行わなければダウンロード時間は増大する。また、本実験ではシードは最もシードからピースをダウンロードしたピアへ優先的にアップロードを行う。ピアの参加のタイミングは、到着時間間隔 20 秒で各 AS 最大 10 人まで参加する。

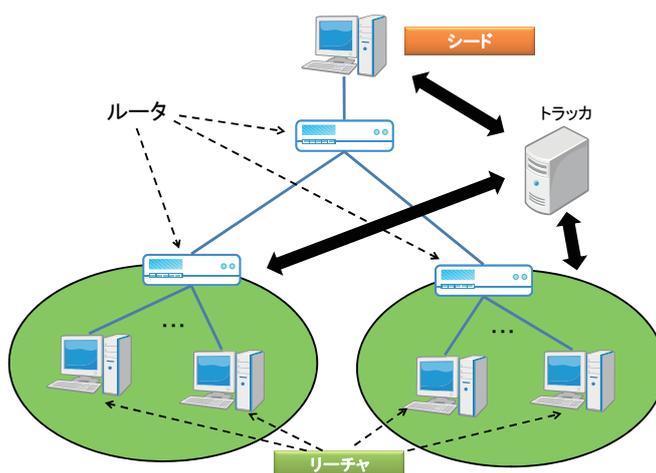


図 9 実験トポロジ

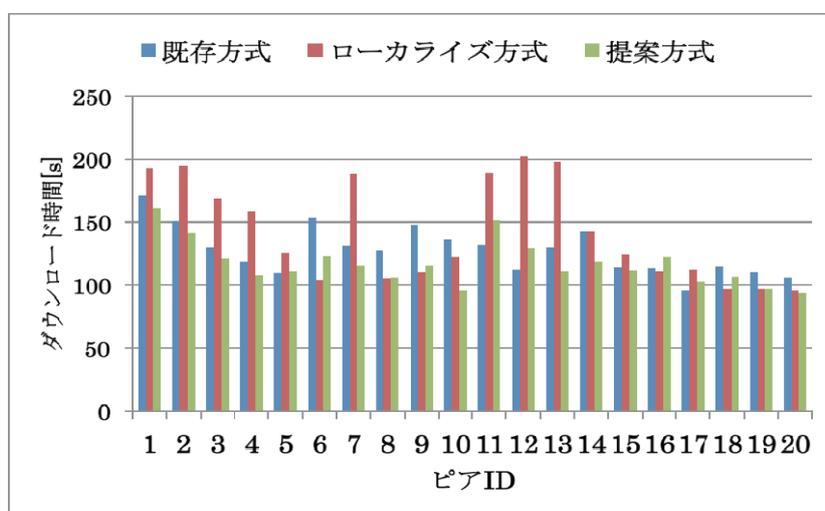


図 10 各ピアのダウンロード時間

図 10に参加時間 20 秒間隔の各ピアのダウンロード時間を示す. 各ピアは到着時間間隔 20 秒でピア 1 から順に各 AS 最大 10 人まで参加し, ピアはダウンロード完了後すぐに離脱する. ファイルは 100MB のダウンロードを行う. 横軸はダウンロードピア ID (奇数が AS1 のピア, 偶数が AS2 のピア), 縦軸はダウンロード時間を表す.

提案方式は既存方式に比べ全ピアの平均ダウンロード時間が約 8%減少した. 一方ローカライズ方式は既存方式と比べ全ピアの平均ダウンロード時間は約 10%増加している. ローカライズ方式では特にダウンロード序盤のピア (1, 2, 3, 4) や, 中盤のピア (11, 12, 13) でダウンロード時間が増加している. これはスウォーム内のピアが少ない段階でもピースの持ち合いに関係なく AS の内側のピアを優先的に選択しようとするため, ダウンロード効率が悪くなっているためだと考えられる.

図 11~図 13にそれぞれ既存方式, ローカライズ方式, 提案方式のそれぞれの AS, スウォーム全体でのピース取得率の時間経過を示す. ピアの離脱の度にピース取得率が減少する. 特にローカライズ方式では, ピースを取得できる範囲が限定されるため, ピース取得率の低下が大きい. しかし, 提案方式では, スウォーム全体に加え, それぞれの AS でもピースがまんべんなく拡散するようにしているため, ピース取得率の低下が小さく, 安定したダウンロードが実現されており, 結果的にダウンロード時間の短縮につながっている.

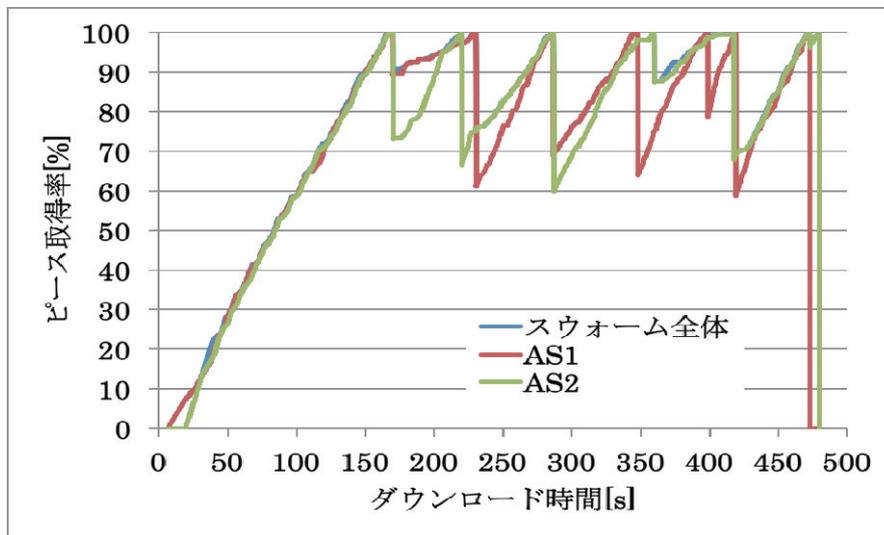


図 11 ピース取得率 (既存方式)

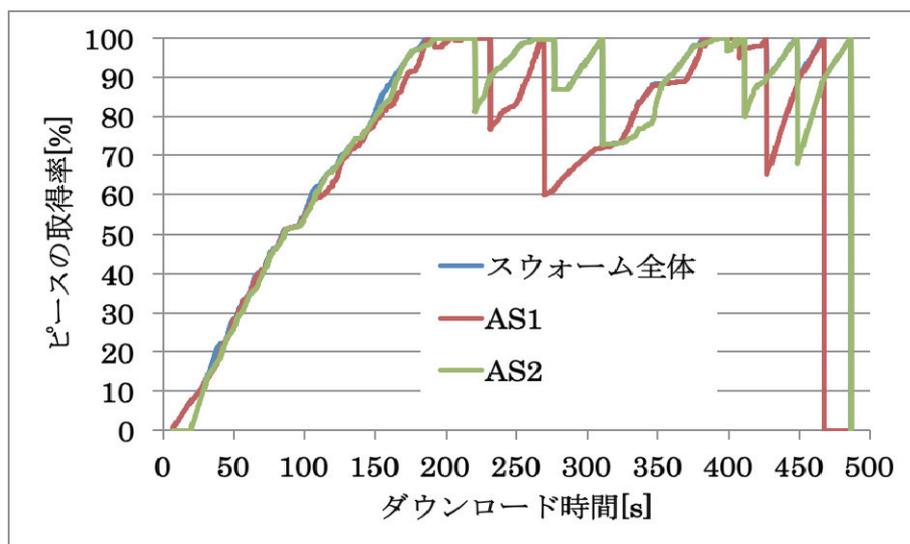


図 12 ピース取得率 (ローカライズ方式)

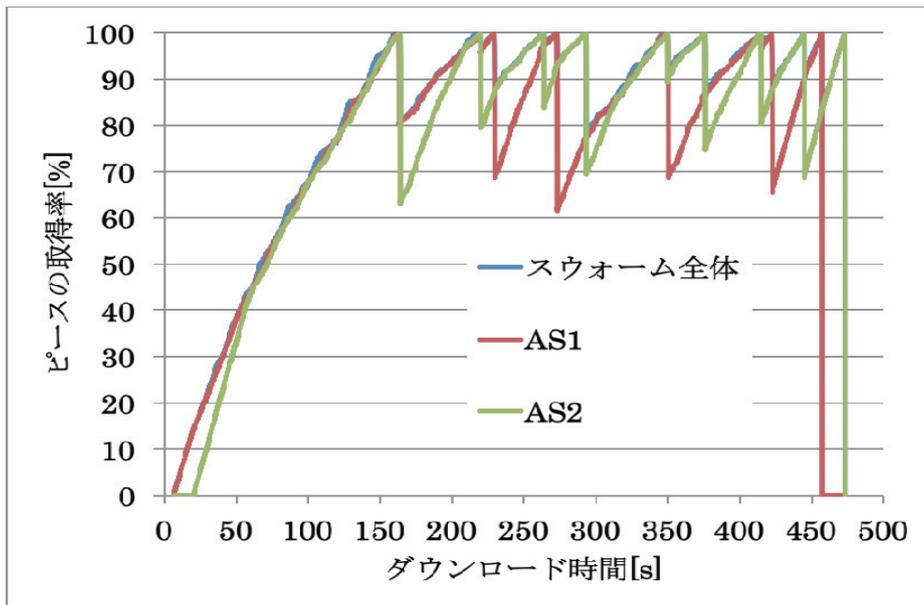


図 13 ピース取得率 (提案方式)

図 14にクロス AS トラフィックの累積を示す。提案方式は、通信範囲を制限したローカライズ方式と同等のクロス AS トラフィック量でありながら、ダウンロード速度を改善していることがわかる。

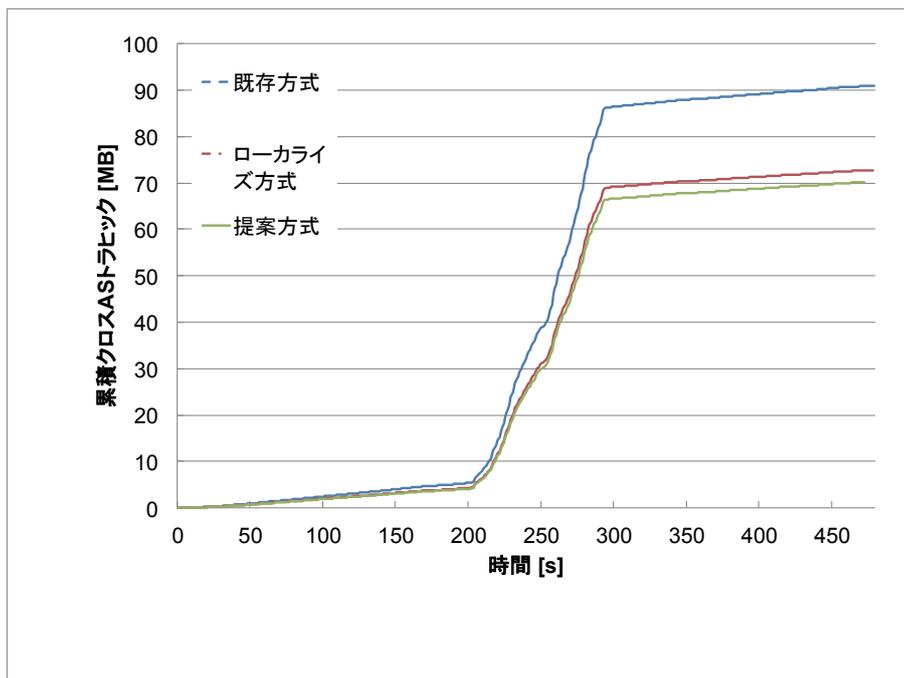


図 14 クロス AS トラフィック

2-7 まとめ

本研究ではピースの塊を考慮した BitTorrent における高速ダウンロード方式を提案した。BitTorrent に用いられるファイルを分割したピースをいくつか束ねた塊として扱いピース選択戦略、ピア選択戦略の戦略を立てやすくし、ダウンロードの効率化を図った。

ピース選択戦略では Random-First を排除し、ピアはトラッカで管理されている AS ごとの塊の記録とスウォーム全体の塊の記録を参照しピースの選択を行うように修正した。ダウンロード序盤はスウォーム全体に塊が分散されるような塊を優先的に選択し、スウォーム全体に塊が行き渡った後に自分の AS に不足している塊を Rarest-First 戦略によってダウンロードすることでスウォーム全体そして、各 AS でピースが上手く分

散するような戦略に変えることでピース選択戦略の効率を上げた。また、自分の所望の同じピースを所持しているピアが複数いた場合に、既存の方式では所持している複数のピア全員に interested メッセージを送信し、一番速く非チョークをしてくれた相手からダウンロードを行っていたが、提案方式では自分に近いピア（同じ AS 内のピア）だけに interested メッセージを送信することで、近くからピースのダウンロードを行うことが可能となり、ピースのローカライゼーションを実現した。

ピア選択戦略では楽天的非チョーク戦略によるランダムなピア選択を廃止し、代わりに塊を利用した戦略を取ることで、AS 内のピースの重複を減らし、さらにローカライズを行うことでダウンロードにおける遅延の回避や冗長トラヒックの抑制を実現させた。初めは AS 内や AS 外のピアに関係なく、自分の所持していない塊を持っているピア、自分の AS に無い塊を持っているピアと優先的に接続するように戦略をとり、AS 内でファイルを構成する全ての塊が集まった後、ローカライズが働くような戦略に切り替えることで AS 間トラヒックの抑制を実現した。

評価実験では、リーチャがダウンロード完了後シードとしてスウォームに残る場合と、ダウンロード完了後スウォームから即離脱する場合についての評価を行った。提案方式は既存方式に比べ、ダウンロード完了時間を減少させ、AS 間トラヒック量を抑制させることができた。

3 BitTorrent における tit-for-tat 戦略の改良による AS 間トラヒック削減

3-1 はじめに

前節では、ピースの拡散とローカライゼーションの関係に着目したが、本説では、ローカライゼーションを実現する別のアプローチとして、tit-for-tat 戦略の改良を行う。tit-for-tat 戦略の改善により、ピアの選択を自身の所属する AS と同じ AS に属するピアに優先的に接続することで、AS 間の冗長トラヒックを削減するローカライゼーションを実現し、AS 間トラヒックの削減を目指す。また、既存方式ではローカライゼーションの働きによって、他 AS のピアとの通信が抑制され、自 AS のピアとの通信が促進されるが、自 AS でピースが取得できない場合に、ファイルのダウンロードができなくなる問題点がある。提案方式では、自 AS でダウンロードができない場合に状況に応じて他 AS とも通信を行うため、先の問題を回避できるようにする。

3-2 ピア選択戦略の改良（提案方式 1）

ピア選択戦略では、前述したように、相手からのダウンロード量のみでピアを選択することによって地理的に遠いピアと接続することを防ぐため、自分と同じ AS に属しているピアを優先的に接続するようにピア選択戦略に変更を加える。これにより、同じ AS に属するピアとの接続機会が増え、異なる AS に属するピアとの接続機会が減るため、結果として AS 間トラヒックを削減することができる。

通常方式で使われている tit-for-tat 戦略では、自分に対するアップロード量が多いピアを非チョークしていたが、それでは、AS 外のピアに対する非チョークが増えてしまい、AS 間トラヒックの増加を招いてしまう。そこで、提案方式 1 では、tit-for-tat 戦略で直近 20 秒の相手からのダウンロード量を計算する際、自分と同じ AS に属しているピアからのダウンロード量のみを計算し、AS 外のピアからのダウンロード量は計算を行わず相手からのダウンロード量をゼロバイトとする。これにより、後のダウンロード量を大きい順に並べて上位 3 ピアを非チョークする際、AS 外のピアからのダウンロード量はゼロバイトであるため、上位 3 位に入りづらくなり、他の AS のピアを非チョークする割合を減らすことができるため、AS 外のピアとの通信が減り、結果として AS 間トラヒックを削減することができる。図 15 に、提案方式における tit-for-tat 戦略のフローチャートを示す。

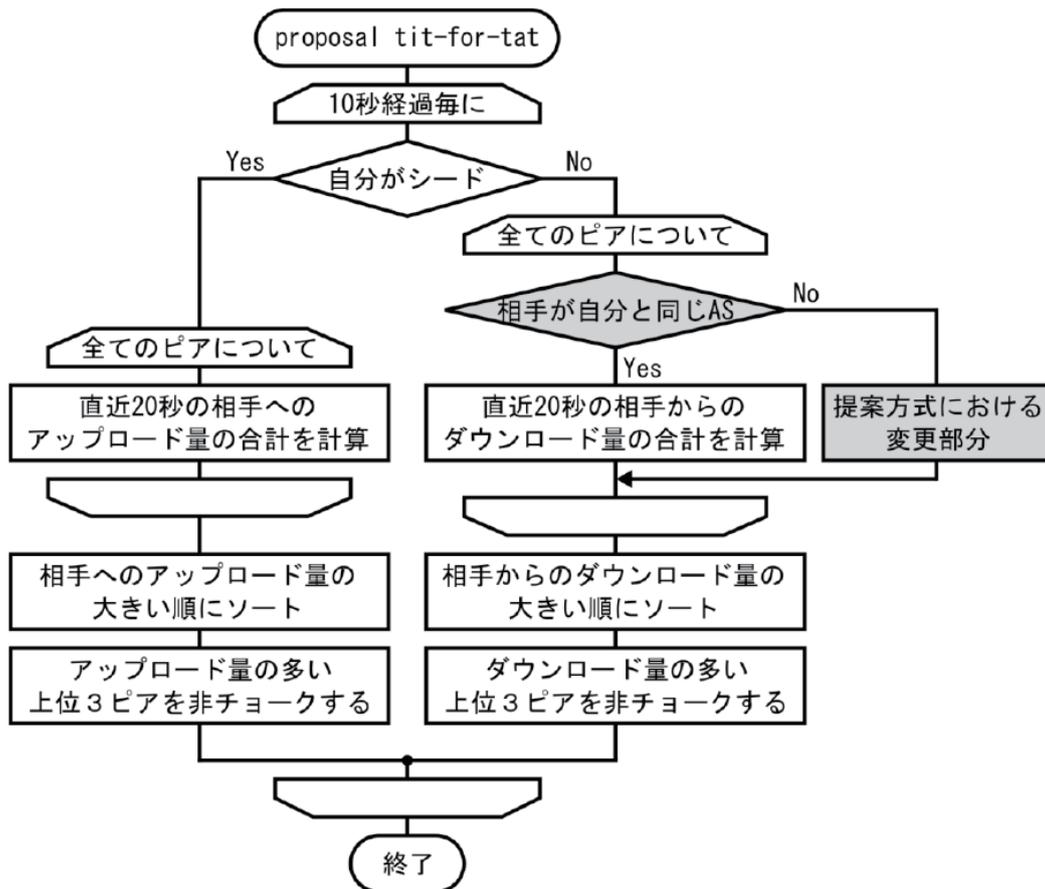


図 15 ピア選択戦略

3-2 ピア選択戦略の改良（提案方式 2）

前節の提案方式では、tit-for-tat 戦略において、各ピアからのアップロード量の総和を計算する際に、他 AS のピアからのアップロード量をゼロとして計算することで自 AS のピアとの通信を促し、ローカライゼーションを実現していた。しかし、この方法では、自 AS のピアからのダウンロードを一度も行っていない場合に、他 AS のピアの中に自分に対するアップロード量が多いピアがいたとしても、すべて同じアップロード量ゼロで非チョークを行ってしまい、ダウンロードの効率が低下してしまうと考えられる。そこで、提案方式 2 では、他 AS のピアからのアップロード量をゼロではなく 1/4 倍にすることで、非チョーク対象となる上位 3 ピアに自 AS のピアを優先させながら、他 AS のピアのアップロード量にも順位をつけ、tit-for-tat 戦略を働かせることで効率を高めるものである。

3-2 ピア選択戦略の改良（提案方式 3）

提案方式 3 では、他 AS のピアからのダウンロード量を次の式で定義する。

$$\text{他 AS ピアからのダウンロード量} = \frac{\text{そのピアからのダウンロード量}}{\text{すべてのピアからのダウンロード量}}$$

他 AS のピアからのダウンロード量が直近 20 秒間の自身の全ダウンロード量に対する割合を計算し、その割合をかける方式である。そうすることで、自 AS のピアからのダウンロード量が少なければ、ダウンロード量が多い他 AS のピアを非チョークする。tit-for-tat 戦略によって非チョークされやすい相手を選ばれるので、ダウンロード効率は良くなると考えられる。

3-3 評価実験

第 2 節と同様の環境で実験を行った。

シナリオ 1 (リーチャはダウンロード完了後離脱しない) :

図 16から提案方式と通常方式ではダウンロード完了時間にあまり差が生じていないことがわかる。これは、ローカライズ方式、提案方式共において、ピア選択戦略を変更したが、ピース選択戦略は変更を加えていないため、接続相手が同じ AS のピアになることによって AS 間トラヒックは減るが、取得するピースの違いによる拡散の効率には変化がないためであると考えられる。

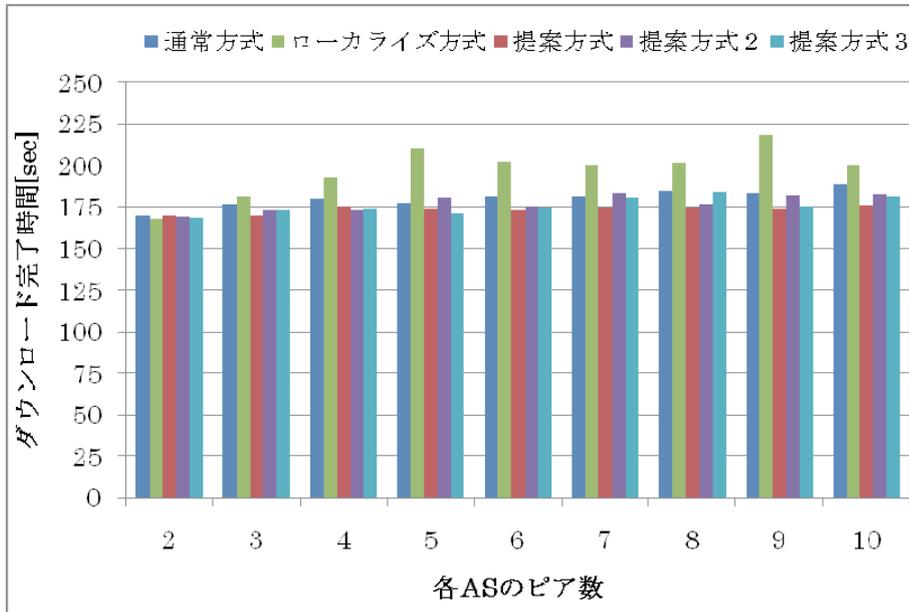


図 16 ダウンロード完了時間 (シナリオ 1)

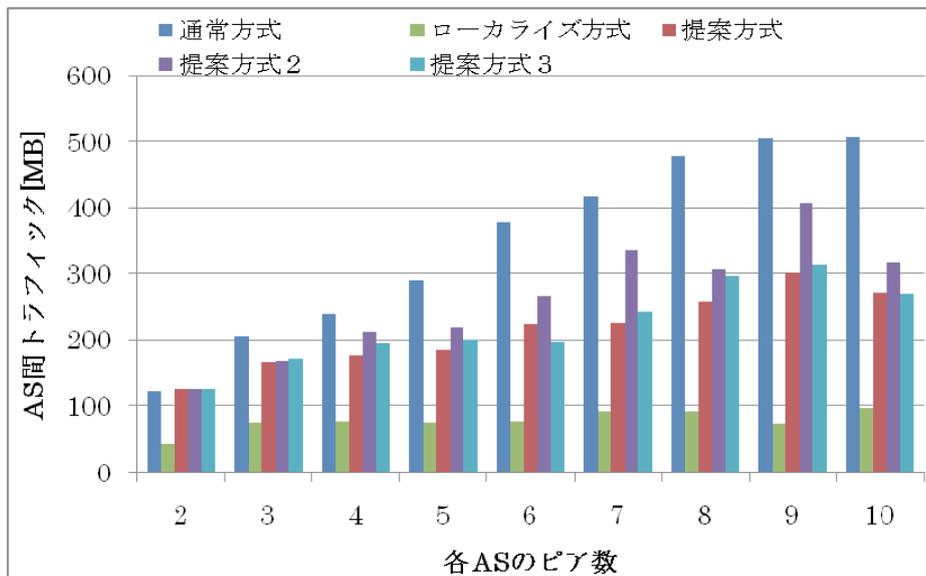


図 17 AS 間トラヒック (シナリオ 1)

図 17にシナリオ 1における各 AS ピア数を増加させた場合の AS 間トラヒックの変化を示す。ローカライズ方式と提案方式では、通常方式よりもトラヒックを抑制させることが読み取れる。通常方式とのトラヒック量の差は、楽観的非チョーク戦略によるランダムなピア選択を排除し、ローカライズを意識し、同じ AS のピアを優先的に選択したことによるものと考えられる。ローカライズ方式と提案方式では、各 AS のピア数が増えても、AS 間トラヒックの増加は少ししか増えない。これは、ピアの数が増えることによって、同じ AS に属するピアの数も増えるため、AS 内でダウンロードする機会が増えるからである。

シナリオ 2（リーチャはダウンロード完了後、即離脱）：

図 18にシナリオ 2における各 AS のピア数を増加させた場合のダウンロード完了時間の変化を示す。シナリオ 2でもシナリオ 1 同様に提案方式は、通常方式と比べ、ダウンロード完了時間がほとんど変わらないことがわかる。これは、シナリオ 1 の場合と同様に、ピース拡散による効率の変化が無いためである。ローカライズ方式では、ダウンロードを完了したリーチャがすぐに離脱してしまうため、残されたリーチャがダウンロードする機会が減ってしまい、ダウンロード完了時間が長くなっている。

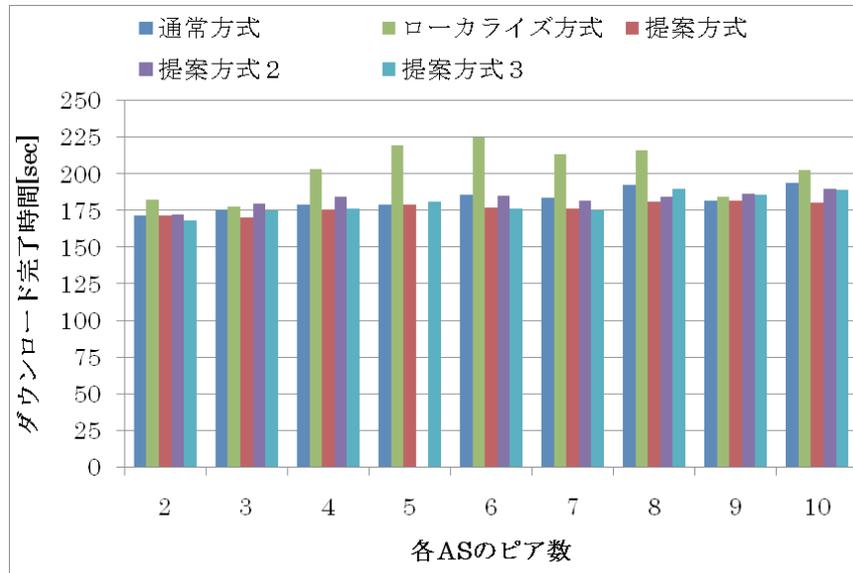


図 18 ダウンロード完了時間（シナリオ 2）

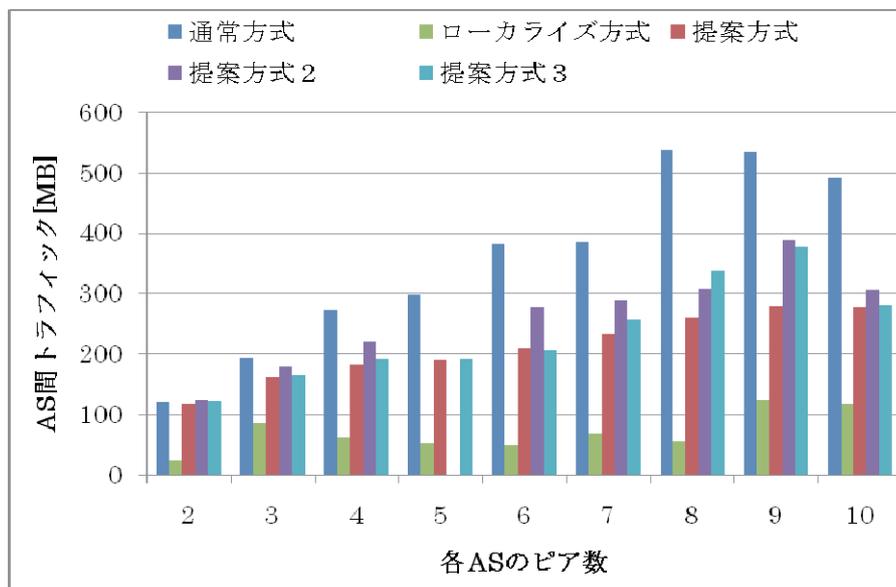


図 19 AS 間トラフィック（シナリオ 2）

図 19にシナリオ 2における各 AS ピア数を増加させた場合の AS 間トラフィックの変化を示す。シナリオ 1 同様に提案方式は通常方式よりトラフィック量が減少し、通常方におけるトラフィック量から 40%程度を抑えることができた。提案方式のピア選択戦略において、同じ AS のピアと優先的に接続するようになったため、AS 間トラフィックを抑制することができたと考えられる。この場合でも、ローカライズ方式では、INTEREST メッセージの送信範囲の制限によって AS 間トラフィックを大幅に削減することができている。

図 20にシナリオ 3 での通常方式と提案方式のダウンロード完了時間を示す。このシナリオでは、各ピア

は20秒毎に各ASに交互に参加するようになっていて、そして、ピアはダウンロード完了後すぐに離脱する。ファイルは100MBのダウンロードを行う。横軸は各ASのピア数、縦軸はダウンロード完了時間を示す。

シナリオ3における、各ピアのダウンロード完了時間は、各ASのピア数が増えるほど短くなっている。これは、一定間隔でピアが参加するため、その間にピースがスウォーム内に広がり、後から入ったピアがピースをダウンロードしやすくなるためである。全体的にシナリオ1、シナリオ2と比べてダウンロード完了時間が短くなっているのもそのためだと考えられる。

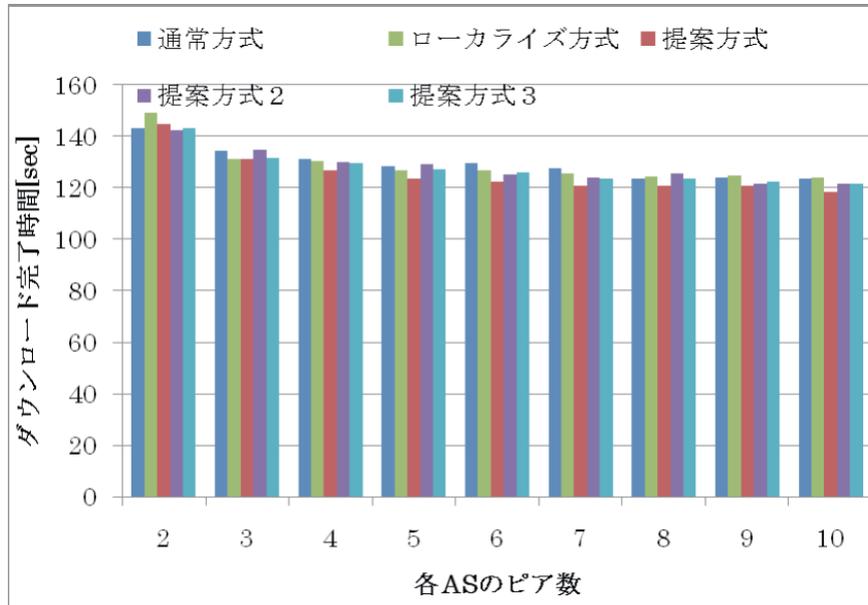


図 20 ダウンロード完了時間 (シナリオ3)

図 21にシナリオ3でのAS間トラフィックを示す。各ピアは到着時間間隔20秒で各AS最大10人まで参加し、ピアはダウンロード完了後すぐに離脱する。ファイルは100MBのダウンロードを行う。横軸は各ASのピア数、縦軸はAS間トラフィックである。シナリオ3では、シナリオ1、シナリオ2に比べ、AS間トラフィックの削減が少なくなっている。この原因は、シナリオ3では、後から入ったピアは前に入っていたピアからピースをダウンロードしやすくなっている。しかし、このシナリオでは、参加するピアはAS1とAS2に交互に入っていくため、後から入ったピアは、もう一方のASのピアからダウンロードすることが多くなり、AS間トラフィックの削減が少なくなっている。

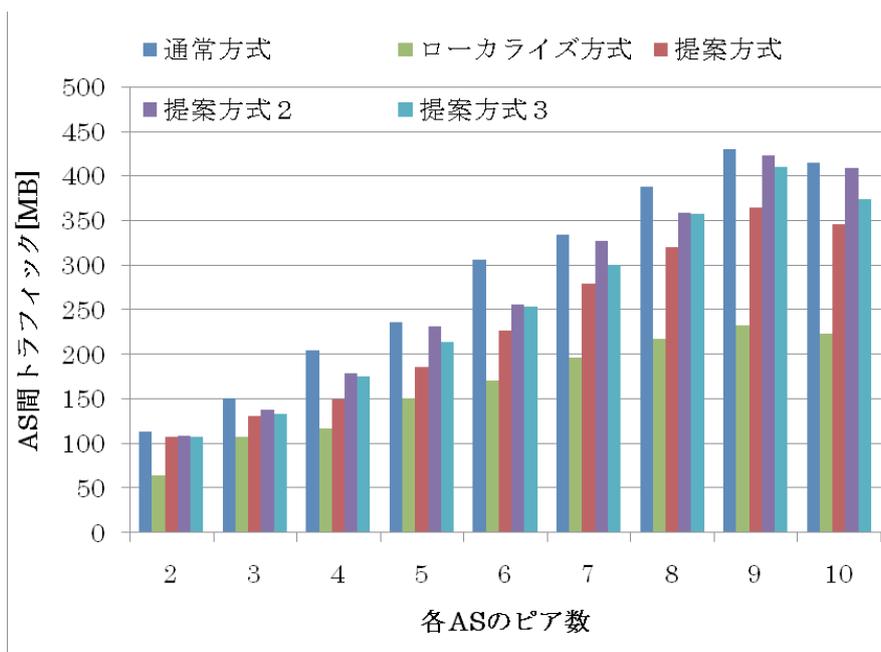


図 21 AS 間トラフィック (シナリオ 3)

3-4 まとめ

本研究ではコンテンツ配信アプリケーションである BitTorrent において、tit-for-tat 戦略のアルゴリズムを改良することで AS 間トラフィックを削減する方式を提案した。非チョークアルゴリズムにおいて、通常方式では、相手からのダウンロード量のみで非チョークを行っていたが、その際、自分と同じ AS に属するピアを優先的に非チョークするように変更することで、他の AS のピアへのアップロード機会が減少し、AS 間トラフィックの増加を抑えることができた。評価実験により、tit-for-tat 戦略の改良によって、ローカライゼーションを実現し、AS 間トラフィックを抑えることができたことがわかった。しかし、ピアの参加タイミングによっては、AS 間トラフィックをあまり抑えることができなかった。シナリオ 1 (リーチャがダウンロード完了後シーダとしてスウォームに残る場合) では、提案方式は通常方式からダウンロード完了時間をほとんど変化させることなく、AS 間トラフィック量を抑制させることができた。シナリオ 2 (リーチャがダウンロード完了後スウォームから即離脱する場合) でも、同様にダウンロード完了時間を増やすことなく AS 間トラフィックを削減することができた。シナリオ 3 では、遅れて入るピアは、すでにスウォーム内にピアが広まっている状態からダウンロードが開始するため、ダウンロード完了時間が短くなる。しかし、このシナリオにおいては、ピアの参加する条件が各々の AS に交互に参加することになっていたため、他の AS のピアと接続する機会が増えてしまい、AS 間トラフィックはあまり抑えることができなかった。

4 まとめ

本調査研究では、ローカライゼーションとピースの拡散を考慮することにより、AS 間トラフィックを減少させながら、ダウンロード時間を短縮することに成功した。さらに、tit-for-tat 戦略を改善することによっても、ローカライゼーションを実現できることも示した。

今後の課題として、提案方式は、トラッカでピースの拡散状況を管理していたので、これを自律分散管理することや、匿名ネットワークである Tor への適応[3]などがある。

【参考文献】

- [1] D.R. Choffnes and F.E.Bustamante. Taming the torrent: A Practical Approach to Reducing Cross-ISP Traffic. In: Peer-to-Peer Systems, In Proc. of ACM SIGCOMM '08, pp.363-374, 2008.

- [2] Michael Piatek, Harsha .Madhyastha, John P.John, Arvind Krishnamurthy, Thomas Anderson, Pitfalls for ISP-friendly P2P design, Proc. of the Eighth ACM Workshop on Hot Topics in Networks(HotNets-VIII), pp.1-6, 2009.
- [3] Timothy Kale, Satoshi Ohzahata, Toshihiko Kato, Improving Tor Circuit Performance with Guard Relay Nodes, IEICE 総合大会, BS-3-14, 2012.

〈 発 表 資 料 〉

題 名	掲載誌・学会名等	発表年月
Promoting Peer Cooperation for Efficient Piece Diffusion under Localization in BitTorrent	IEICE IN 研究会	2011 年 6 月
Improving Tor Circuit Performance with Guard Relay Nodes	IEICE 総合大会	2012 年 3 月