

推薦論文

ブロックのレアリティを考慮した 効率的な P2P ファイル共有手法

松本 敬^{†1} 遠藤 伶^{†1} 重野 寛^{†2}

P2P ファイル共有において、レアリティが高く入手の難しいブロックが発生し、ブロック収集効率が下がるブロックのレアリティ問題が存在する。ブロックとは、共有するファイルをあらかじめ決められたサイズに分割した断片のことである。そこで、本論文ではブロック収集効率を上げるためにブロックのレアリティを考慮したブロックを効率的に分散させる P2P ファイル共有手法 CAS の提案を行う。CAS ではレアリティ問題の原因であるブロックの分散速度とピア離脱の 2 点に対処することで、ブロックを収集するための効率をあげる。さらに、シミュレーション評価を行い、ネットワーク内のピア数に変動がない場合にネットワーク内の全ピアがファイル復元に要する時間が、CAS は既存手法と比べ 60% に短縮されることを示した。

Efficient P2P File Sharing Method Considering Rarity of Blocks

KEI MATSUMOTO,^{†1} REI ENDO^{†1} and HIROSHI SHIGENO^{†2}

This paper discusses a rarity problem of blocks in P2P file sharing that de-generates the blocks collection efficiency of peer by the blocks that have high rarity and thus difficult to obtain it. Blocks are the divided fragments of shared files as which size is decided previously. The aim of this proposal is to improve the blocks collection efficiency of peer in P2P. We propose CAS which is a P2P file sharing method for efficient distribution of blocks that considers the rarity of blocks. In the proposal, CAS deals both the dispersion speed of blocks and the peer departure to improve the blocks collection efficiency. Moreover, we show through simulations when number of peer is stable that CAS reduces file reconstruction time of all peers in network to 60 percent than traditional method.

1. はじめに

P2P 技術を用いた代表的なサービスとして P2P ファイル共有があげられる。P2P ファイル共有では、対等な立場でネットワークに接続するピアによりファイルを分割した断片が交換される。この断片のことをブロックと呼ぶ。ピア間でブロック交換をし、全ブロックを集めて元ファイルを復元させることでファイル共有を行う。本論文では便宜上あるピアが他ピアへブロックを渡すことをアップロード、他ピアからブロックを受け取ることをダウンロードと呼ぶ。また P2P 技術を使用しているため、ピアはファイルのアップロード帯域をネットワークに提供していると考えることができる。ピアの増加に対応して柔軟にネットワークリソースを向上させられるためスケラビリティがあるのが特長である。

P2P ファイル共有サービスにおける問題の 1 つに、ブロックのダウンロードだけを行いアップロードを行わないピアの free-riding 問題¹⁾が存在する。このようなピアを悪意あるピアと呼ぶ。悪意あるピアは、ネットワークにリソースを提供しない。リソースが提供されないとブロック交換の効率が落ちるため、悪意あるピアが存在すると P2P ファイル共有の性能は低下する。代表的な P2P ファイル共有ソフトである BitTorrent²⁾では、free-riding 問題の対策手法としてブロック交換手法に TFT (Tit-for-Tat)³⁾⁻⁵⁾を用いている。TFT はブロック交換を行う 1 対 1 のピア間にダウンロード数に関する制約を設けている。すなわち、交換相手からブロックをダウンロードできる数をその交換相手へアップロードしたブロック数以下で制約する。実際の環境ではアップロード帯域とダウンロード帯域が非対称な環境が存在し、TFT ではアップロード帯域が小さい場合に使用可能なダウンロード帯域が制限される。加えて、新規参加ピアなどの交換相手へ提供できるブロックを所持しない場合にブロック交換が成立しない。これらの場合、本来使用可能なダウンロード帯域が無駄となりブロックを収集する効率が低下する。

また、P2P ファイル共有サービスでは、特定のブロックが入手できないためファイルの復元が遅くなる問題が存在する。ブロックは、ブロック交換を行う過程でネットワークへ分散される。そして、ブロックによりネットワークへの分散の仕方にばらつきがあるため、

^{†1} 慶應義塾大学大学院理工学研究科
Graduate School of Science and Technology, Keio University

^{†2} 慶應義塾大学理工学部
Faculty of Science and Technology, Keio University

本論文の内容は 2009 年 3 月のマルチメディア通信と分散処理研究会にて報告され、同研究会主催により情報処理学会論文誌ジャーナルへの掲載が推薦された論文である。

ブロックによってネットワークに存在する数が異なる。分散した結果、あまり普及せずネットワーク内に存在する数が少ないブロックは、他のブロックに比べ入手困難となる。本論文では、この問題のことをブロックのレアリティ問題と呼び、ネットワーク内に存在する数が少ない入手困難なブロックのことをレアリティが高いブロックと呼ぶ。

レアリティが高いブロックの発生する原因は、特定のブロックの分散速度が遅い点にある。ブロックの分散速度とは、ブロックがネットワークへ広がる速さである。ブロックの分散速度が遅いと、そのブロックがネットワーク全体へ行き渡るまで時間がかかる。もう 1 つの原因は、レアリティの高いブロックを入手したピアが短時間のうちに離脱することである。高レアリティなブロックの入手は、低レアリティなブロックと比べ時間がかかる傾向にある。結果、ピアが高レアリティなブロックを入手してからネットワークを離脱するまでの時間は短く、高レアリティなブロックほどネットワークへ提供される時間が短くなる。そのため、レアリティが高いブロックはネットワークへ分散し難く入手困難な状態が持続する。これらから、効率の良い P2P ファイル共有ではブロックのレアリティ問題を考慮する必要があるが既存の BitTorrent では考慮されていない。

本論文では、ブロックのレアリティ問題を考慮したブロックを効率的に分散させる P2P ファイル共有手法 CAS (Carrot and Stick) を提案する。TFT ではブロック交換を行う 1 対 1 のピア間へダウンロード数に関する制約を設けているのに対し、CAS では過去に他ピアへアップロードしたブロック数情報を用いることでブロック交換を行う 1 対多のピア間に制約を設ける。また、制約条件である必須アップロード数は、各ピアの入手ブロック数に応じて動的に変化させる。必須アップロード数を動的に変化させることで、レアリティ問題の原因であるブロックの分散速度とピア離脱を改善する。必須アップロード数とは、そのピアにとって新しいブロックをダウンロードするために必要な他ピアに対するアップロード数である。この新規ブロックのダウンロードの際に、CAS はファイル分割数に対するピアが所持するブロック数の割合 (所持ブロック率) の低い間は必須アップロード数を少なくし、所持ブロック率が高くなると必須アップロード数を多くする。これにより、レアリティが高いブロックを長く提供できるようになりレアリティ問題が抑制される。CAS によりブロックのレアリティ問題が考慮され、ブロックを効率的に収集することが可能となる。

2 章において関連研究として BitTorrent の既存手法と問題点、そして本提案で扱うブロックのレアリティ問題の詳細を説明する。3 章でブロックのレアリティ問題を考慮したブロックを効率的に分散させる P2P ファイル共有手法 CAS を提案する。4 章でシミュレーション評価による CAS のブロック収集効率における有用性を示す。最後に 5 章で結論を述べる。

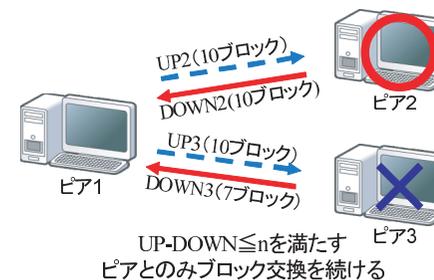


図 1 Tit-for-Tat の例 ($n = 2$ の場合)
Fig. 1 Example of Tit-for-Tat ($n = 2$).

2. 関連研究とブロックのレアリティ問題

関連研究として、代表的な P2P ファイル共有ソフト BitTorrent のブロック交換で使用されている既存手法の TFT (Tit-for-Tat), および BitTorrent 上で悪意ある行動をとる BitThief について説明する。そして、ブロックのレアリティ問題の詳細について説明する。

2.1 関連研究

TFT は、ブロック交換を行う 1 対 1 のピア間へダウンロードに関する制約を設けている。交換相手からブロックをダウンロードできる数とその交換相手へアップロードしたブロック数に依存する制約である。この制約により、ファイル復元に必要なブロックをダウンロードするために、アップロードを強制するため free-riding 問題に対処できる。

図 1 に TFT によるブロック交換の例を示す。図 1 は、ピア 1 がピア 2 とピア 3 という 2 ピアを相手にブロック交換を行っている様子である。上記の TFT における制約は、各ピアがブロック交換を行う際に以下の式に従うことで実現している⁶⁾。

$$UP - DOWN \leq n \quad (1)$$

ブロック交換を行う各ピアは、ブロック交換の相手に対して式 (1) の計算を行い、式を満たしているピアのみとブロック交換を続ける。式 (1) の n はあらかじめ設定しておく固定値で、ブロック交換の相手へいくつまで多くブロックのアップロードを許容するかの値である。本論文では便宜上この n のことを許容値と呼ぶ。図 1 は許容値 n を 2 に設定した場合の例で、ピア 1 はピア 2 とはブロック交換を続けるが、ピア 3 とのブロック交換は中断している。これは、ピア 1 が各ピアに対して式 (1) を計算して条件を満たしているかいないかを判断した結果である。ピア 2 との計算結果 $UP2(10) - DOWN2(10) = 0$ は許容値 2 以

下と式 (1) の条件を満たしているためブロック交換を継続している。しかし、ピア 3 との計算結果 $UP3(10) - DOWN3(7) = 3$ は許容値 2 を超えており式 (1) の条件を満たしていないためピア 1 はブロック交換を中断する。ピア 1 がブロック交換を中断すると、ピア 3 がいくらピア 1 に対してブロックのアップロードを行っても新規ブロックを入手することはできない。するとピア 3 も評価式が満たされなくなり、ピア 3 もブロック交換を中断することとなる。結果として、両ピアともブロック交換を中断することになる。ブロック交換相手へ n 個までブロックのアップロードを許容するのは、ある程度の許容幅がないとブロックを所持していないネットワークへの新規参加ピアがブロックを 1 つもダウンロードできなくなるからである。ピアがブロックのダウンロードを続けるためには、ブロック交換相手との間で式 (1) の条件を満たすためにブロックをアップロードし続け、ブロック交換相手に対する UP 値を増加させる必要がある。これらから、他ピアへブロックをアップロードする動機付けが行われており、free-riding 問題の対策手法として用いられている。

しかし、TFT はピアの環境によってブロックを収集する効率が変わってしまう問題がある。TFT はブロックのダウンロード数がアップロード数に依存している。ピアのダウンロード帯域とアップロード帯域の大きさが同程度の場合なら問題ないが、実際の環境ではダウンロード帯域とアップロード帯域の大きさが同程度の場合は少なく、ダウンロード帯域に対してアップロード帯域が小さいことが多い。ほかに、ネットワークに参加したばかりのピアなどの交換相手へ提供できるブロックを所持しない場合はブロック交換がうまく成立しない。これらの場合には、本来使用可能なダウンロード帯域を十分に使用できず無駄になる。後に述べるブロックのレアリティ問題の観点からすると、ダウンロード帯域を十分に利用できないのはファイルの復元が遅くなる原因となる。

BitThief⁷⁾ では、他ピアへブロックのアップロードなしにダウンロードできるように BitTorrent を改造している。つまり、BitTorrent 上で free-riding を行うためのソフトウェアである。BitTorrent における既存手法である TFT では、ブロックの交換相手からダウンロードできるブロック数がその交換相手へアップロードしたブロック数に依存する制約を設けている。しかし、ブロック未所持のピアのためにブロックをアップロードせずにダウンロードが可能な許容幅が存在する。BitThief では、この許容幅だけを使いダウンロードを行う。具体的には、ブロックの交換相手へつねにアップロードするブロックを持たないピアに偽装し続けることで、ブロックのアップロードなしにダウンロードだけを行う。

しかし、BitThief を使用したところでファイルの復元時間が早くなるわけではない⁸⁾。BitThief を使用しているピアは、P2P ファイル共有において悪意あるピアと同等の動きを

する。これは、ブロックのアップロードを行わないピアはネットワークにリソースを提供しないからである。結果として、P2P ファイル共有の性能が落ちてしまう。

2.2 ブロックのレアリティ問題

ブロックのレアリティ問題とは、ネットワークへの普及率が低く入手が困難なブロックにより、ブロック収集効率が下がるという問題である。ブロックは、ピア間で交換が行われるうちにネットワークへ分散する。ピアのブロック交換では、まず最初にピアが隣人ピアとお互いの所持ブロック情報を比較する。もし未所持なブロックを発見すればダウンロードの要求を行い相手からブロックのコピーをダウンロードし、逆に相手からダウンロード要求がきた場合にはブロックのコピーを相手にアップロードする。ただし、このブロック交換では各ピアの帯域の状態など制限条件が存在する。そのため、制限に影響されずにネットワークへ広く分散するブロックや制限の影響を受け分散し難いブロックが発生する。結果として、ブロックにより分散の仕方にばらつきが発生し、ネットワークへの普及率が異なる状況となる。分散した結果ネットワークへの普及率が低くなってしまったブロックは、他のブロックと比べて入手が困難となる。入手困難なブロックが発生すると、そのブロックを入手できないためにピアのファイル復元が遅くなり、最悪の場合にはファイルが復元できなくなる。本論文では、この問題をブロックのレアリティ問題と呼び、入手困難なブロックをレアリティが高いブロックと呼ぶ。レアリティが高いブロックの発生原因は、上記で説明した各ブロックのネットワークへの分散速度に差が生じることによる普及率のばらつきと、ピアがネットワークから離脱する際にそのピアが所持していたブロックが消失することである。

以下で、ブロックのレアリティ問題に関するブロックの分散速度とネットワークからのピア離脱の 2 点について説明する。

2.2.1 ブロックの分散速度

ブロックの分散速度とは、ブロックがネットワークへ広がる速さである。ブロックの分散速度は、そのブロックを所持するピアの性能に依存する。ピアの性能とは、各ピアが使用可能なアップロード帯域とダウンロード帯域のことである。ピアの性能が高いほどブロックがネットワークへ広がる速度が速くなり、ネットワークへ普及しやすい。そのため、性能が高いピアを中心に広がったブロックは入手困難になり難い。逆に、ブロックが性能の低いピアを中心に広がる場合がある。これは、P2P ファイル共有ではピアによるブロック交換が分散的に行われるためである。そのようなブロックは、ネットワークへ広がる速度が遅くなりネットワークへの普及率が上昇し難くなる。その結果、他のブロックと比べネットワーク内に存在する数が少ないために、入手困難なレアリティが高いブロックとなる。TFT のよう

にダウンロード帯域を十分に利用できずにピアの性能が活かせなくなるのはレアリティが高いブロックの発生要因となる。

2.2.2 ピア 離 脱

ピア離脱には、ファイルの復元に完了したピアによる離脱とランダムな離脱が存在する。ファイルの復元に完了したピアによる離脱はファイル入手したピアがネットワーク内にいる理由がなくなるため起き、ランダムなピア離脱は物理的な回線の遮断やマシンの故障などにより起きる。ブロックのレアリティ問題の原因となるピア離脱は、ファイルの復元に完了したピアによる離脱である。レアリティが高いブロックを入手するには時間がかかるため、ファイルを復元してすぐにネットワークから離脱されるとそのピアは短時間しかレアリティが高いブロックを提供しないことになる。その結果、レアリティが高いブロックはネットワークへ分散し難く入手困難な状態が持続する。ランダムなピア離脱もブロックのレアリティが高くなる要因になりうるが、意図して起きる離脱ではないうえにそのピアがレアリティの高いブロックを所持しているとも限らないため本論文でランダムな離脱は対象としない。

3. 提 案

本論文では、P2P ファイル共有においてブロックを収集するための効率が悪くなってしまいうブロックのレアリティ問題に着目し、ブロックを効率的に分散させる P2P ファイル共有手法 Carrot and Stick (CAS) を提案する。ブロックのレアリティ問題の原因は、ブロックの分散速度とネットワークからのピア離脱である。そのため、CAS ではブロックの分散速度とネットワークからのピア離脱の 2 点に対処する。

まず、ブロックの分散速度を速くするために、所持ブロック数が少ないピアのダウンロード帯域を有効活用できるようにする。ダウンロード帯域の有効活用とは、本来使用可能なダウンロード帯域を十分に利用することである。CAS では、所持ブロック数に応じて動的に変化する必須アップロード数をピアに対する制約として設けている。必須アップロード数とは、新規ブロックをダウンロードするために他ピアに対して必要なアップロード数である。この必須アップロード数による制約を所持ブロック数が少ないピアに対して弱め、ダウンロード帯域を有効活用することでブロックの分散速度を速くする。ネットワークからのピア離脱に対しては、必須アップロード数による制約を所持ブロック数が多いピアに対して強めピア離脱を制限する。図 2 に CAS におけるピアの挙動イメージを示す。また、既存手法の TFT では 1 対 1 のピア間での制約を設けていたのに対し、CAS では過去アップロード数

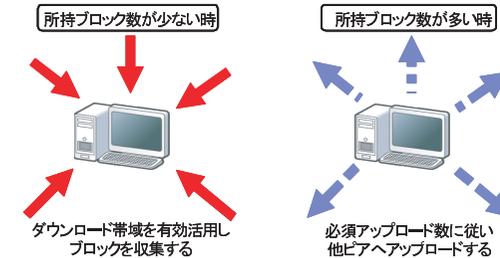


図 2 CAS のイメージ
Fig. 2 Image of CAS.

を用いて必須アップロード数を決定することで、1 対多のピア間での制約としている。過去アップロード数とは、現在までに各ピアが他ピアに対してアップロードを行ったブロックの合計数である。1 対多のピア間での制約にすることで、BitThief のように多くのピアにダウンロード要求すると free-riding ができるのを防止できる。

3.1 ダウンロード帯域の有効活用

P2P ファイル共有では、ブロックの分散速度がブロックのレアリティ問題の原因となっている。CAS では、ピアがダウンロード可能なブロック数を必須アップロード数により制約している。そして、所持ブロック数が少ないピアの必須アップロード数を小さくすることでダウンロード帯域を有効活用できるようにする。その結果、ブロックの分散速度が速くなりブロックのレアリティ問題は抑制される。ブロックがネットワークへ広がるには、そのブロックを所持していないピアへと受け渡される必要がある。そのため、所持ブロック数が少ないピアの存在はブロックの分散速度における影響が大きい。必須アップロード数が小さいと、他ピアへのブロックのアップロード数が少なくてもブロックをダウンロードできる。所持ブロック数の少ないピアが、ブロック交換を行う相手ピアの未所持なブロックを所持している確率は低い。既存手法の TFT のようなダウンロードの制限が状況によって変化しない静的な制約の仕方だと、ネットワークに参加したばかりのピアのブロック交換がうまく成立せずにピアの性能を十分に活かせない。ピアの性能が十分に活かせないとブロックの分散速度は遅くなる。それに対し CAS では、そのようなピアに対する必須アップロード数は小さくダウンロード制限が弱いのでダウンロード帯域を有効活用できる。所持ブロック数の少ないピアがダウンロード帯域を有効活用することで、ブロックはネットワークへ広がりやすくなりブロックの分散速度は速くなる。また、TFT では前に述べたように環境によって本

来使用可能なダウンロード帯域が十分に使用できずに無駄となり、ピアの性能が落ちブロックの分散速度が遅くなる場合がある。しかし、CAS の場合はブロックの分散速度への影響が大きい所持ブロック数の少ないピアはブロックをダウンロードするための必須アップロードの制約が弱いため、ブロックの分散速度が環境に影響され難い。

3.2 ピア離脱の制約

P2P ファイル共有ではネットワークからのピア離脱が起き、ブロックのレアリティ問題の原因となっている。ピア離脱には、ランダムなピア離脱とファイルの復元に完了したピアによる離脱が存在する。本論文ではファイルの復元に完了したピアによる離脱を制約することでブロックのレアリティ問題を抑制する。ファイルの復元に完了するとピアはネットワークから離脱する。そこで、ファイルを復元する前の所持ブロック数が多いピアに対する必須アップロード数を多くすることでピアのネットワークからの離脱を制約する。ただし、ピアにファイルを復元させないわけにはいかないので、CAS では必要最低限の必須アップロード数をファイル分割数、つまりファイルの復元に必要なブロック数とした。

3.3 必須アップロード数

CAS では、必須アップロード数を各ピアの所持ブロック率によって動的に変動させることで、ピアがダウンロードできるブロック数を制限している。新規参加ピアなどの所持ブロック数が少ないピアに対しては必須アップロード数を小さくし、ブロックの分散速度を速めるのに貢献させる。逆に所持ブロック数が多いピアに対しては必須アップロード数を大きくし、ファイル復元によるネットワークからの離脱を制約する。つまり、CAS において満たすべき最低条件は、所持ブロック数が少ないピアには必須アップロード数を少なく、所持ブロック数が多いピアには必須アップロード数を多くすることである。

ピア i の必須アップロード数を N_{upi} とした場合、式 (2) により求める。

$$N_{upi} = S^{x_i} - 1 \quad (0 \leq x_i \leq 1) \quad (2)$$

ここで、 S はファイル分割数、 x_i はピア i の所持ブロック率を表している。式 (2) で指数関数を用いているのは、既存手法と比べ上記の条件をより強く満たし、簡単な式で実現可能だからである。また、ブロックが一定数集まるまで必須アップロード数を 0 とし、ファイル復元寸前の必須アップロードを大きな値に人為的に指定した場合を検討した。しかし、CAS の特性と似た傾向となり大きな差は見られなかった。図 3 に CAS における所持ブロック率と必須アップロード数の関係を示す。

必須アップロード数が S^x ではなく -1 しているのは、 S^x だとブロックを 1 つも所持していない場合にブロックをダウンロードできないからである。

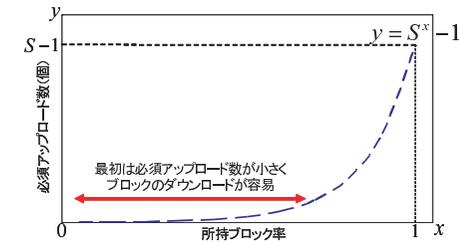


図 3 CAS の特性
Fig. 3 Feature of CAS.

3.4 ブロック交換アルゴリズム詳細

ブロック交換方法以外は既存手法と同じため、ここではブロック交換のアルゴリズムについて説明する。BitTorrent では定期的に隣人リストを更新し、ブロック単位でダウンロードを行う。その結果、隣人ピアの所持ブロックの中に未所持のブロックを見つけた場合、その隣人ピアに対してブロックのダウンロード要求を送信する。もし、複数のファイルが混在している場合には、個別ファイルごとにファイル分割数情報をすべてのピアが保持し、CAS の適用も個別ファイルごとに行う。なお、ピアの目的はすべてのブロックを収集しファイルを復元し入手することなので、ピアは 1 度所持したブロックはネットワークから離脱するまで保持する。ここでは、ダウンロード要求を送信したピアを *Receiver*、*Receiver* からのダウンロード要求を受信したピアを *Sender* とする。

(1) ブロック交換の開始

Receiver は *Sender* に対して、ブロックのダウンロード要求と *Receiver* の所持ブロック数、過去アップロード数情報を送信する。過去アップロード数とは、今までに他ピアに対してアップロードを行ったブロックの合計数である。

(2) 必須アップロード数の計算

Receiver からのダウンロード要求を受け取った *Sender* は、式 (2) から *Receiver* の必須アップロード数を計算する。

(3) アップロードの判断

Sender は、*Receiver* の必須アップロード数と過去アップロード数を比較する。もし過去アップロード数が必須アップロード数より多い場合は、*Receiver* は新規ブロックを入手できる状態と判断する。そして、*Sender* は *Receiver* に対して要求ブロックのアップロードを行う。逆に、過去アップロード数が必須アップロード数より少ない

場合は、Receiver は新規ブロックを入手できない状態と判断する。そして、Sender は Receiver に対して要求ブロックのアップロードを行わない。

4. シミュレーション評価

CAS のブロック収集効率の有用性を示すため、P2P ファイル共有ソフト BitTorrent の既存手法 TFT との比較評価を行った。シミュレーションシナリオは以下とした。

- ブロックを収集しファイル復元を目的としたピアをネットワークへ参加させる。
- シードからネットワーク内のピアへブロックがアップロードされブロック交換が始まる。
- 各ピアが隣人リスト内のどの隣人とブロック交換するかはランダムに決定する。
- 隣人とのブロック交換は Round 単位で行う。
- ブロック交換方法は、TFT は隣人からダウンロードしたブロック数により式 (1) に従う。CAS は各ピアの所持ブロック数と過去アップロード数により式 (2) に従う。
- 全種類のブロックを入手しファイル復元したピアはネットワークから離脱する。
- ネットワーク内の全ピアがファイルを復元した時点で、シミュレーションを終了する。

4.1 シミュレーション条件

実際の環境において、ファイル共有ネットワークに参加するピア数は、ファイルにより異なり様々である。そこで、予備実験としてネットワーク参加ピア数とファイル復元完了時間の関係を測定したが、ピア数のファイル復元完了時間への影響はほとんど見られなかった。そこで本シミュレーションでのネットワークへ参加する総ピア数はシミュレーション時間を考慮し、ネットワークへ途中から参加するピアがない場合は 1,000、いる場合には最大 6,000 とする。また、ネットワークへ途中から参加するピアのことを途中参加ピアと呼ぶ。共有ファイルの分割数、つまりファイル復元に必要なブロック数は 5,000 とする。これは実際のファイルサイズにすると約 2GB に相当する。ファイルサイズ 2GB という値は、実際のファイル共有で扱われるファイルの中でも大きなファイルサイズに分類される動画を想定している。具体的には、1,024 × 768 ピクセルで高画質な 2 時間の圧縮された動画を想定している。ファイルの発信元であるシード数は 1 とし、毎 Round 3 つのピアに対して、3 ブロックずつのアップロードをシミュレーションが終了するまで行う。各ピアの帯域モデルは通常ピアは ADSL、高性能ピアは FTTH とする⁹⁾。通常ピアは Round ごとの最大アップロード数を 3 ブロック、最大ダウンロード数を 10 ブロックとし、アップロード帯域とダウンロード帯域を非対称に設定する。高性能ピアは、最大アップロード数と最大ダウンロード数が 15 ブロックと帯域を対称的にし、ピアの性能を通常ピアより高く設定する。通常ピア

と高性能ピアの存在比率は、帯域が非対称な環境のピアが多くてもブロックを効率的に収集できるか確認するため 2 対 1 とする。各ピアはブロック交換を行うため隣人のリストを持ち、その最大隣人数を 10、定期的に行う隣人の更新周期は 3 Round とする。TFT の式 (1) の許容値 n は広く用いられている 2^{10} とする。

評価項目は、平均ファイル復元時間、ブロックの平均普及時間、悪意あるピアが存在する場合の平均ファイル復元時間とする。平均ファイル復元時間は、ネットワークにピアが参加してからファイルの復元を完了するまでに要する時間の平均である。ブロックの平均普及時間は、ブロックがピアに受け渡され普及するのに要する時間の平均である。悪意あるピアが存在する場合の評価とは、システムをクラックされた際の評価である。CAS ではシステムがクラックされた場合への対処を行っていない。そこで、悪意あるピアが存在する場合の CAS と悪意あるピアが存在しない場合の TFT を比較することで、ブロックのレアリティ問題を考慮することの有用性を示す。本シミュレーションにおける悪意あるピアは、システムをクラックすることで過去アップロード数を多く申告し、アップロードを行わずダウンロードだけ行うピアとする。平均ファイル復元時間の評価は、途中参加ピアがない場合に加え途中参加ピアがいる場合の 2 パターン行う。悪意あるピアが存在する場合の評価は、悪意あるピアの比率が総ピア数の 10%、30%、50%である場合の 3 パターン行う。シミュレーション回数は各 50 回行う。

4.2 シミュレーション結果

CAS と TFT における平均ファイル復元時間、ブロックの平均普及時間、および CAS のネットワーク内に悪意あるピアが存在する場合の平均ファイル復元時間について考察する。

4.2.1 平均ファイル復元時間

図 4 に途中参加ピアがない場合の平均ファイル復元時間の比較を示し、図 5 に途中参加ピアがいる場合の平均ファイル復元時間を示す。ファイル復元率は、全ピアに対するファイルの復元に完了したピアの割合である。

途中参加ピアがない場合は、CAS の方が TFT より全ピアがファイル復元を完了させるのにかかる時間が短い。本シミュレーションでは、TFT のファイル復元率が 1,300 Round 以降から増加しており、平均ファイル復元時間は約 65% のピアがファイル復元を完了させるまでは CAS より早い。しかし、すべてのピアがファイル復元を完了させるのに 2,800 Round かかり、50% のピアがファイル復元を完了させるのに必要な時間 1,400 Round の倍となった。それに対し、CAS のファイル復元率が大きく増加しはじめるのが約 1,600 Round と TFT より遅いが、1,800 Round までにすべてのピアがファイル復元を完了させている。そのた

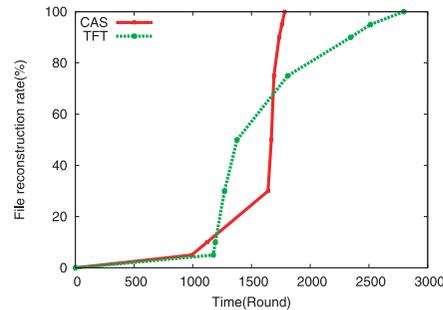


図 4 平均ファイル復元時間の比較

Fig. 4 Compare average file reconstruction time.

め、65%以降までを考慮すると平均ファイル復元時間はCASの方が早い。これは、CASがピア離脱の制約を行うためである。ピア離脱の制約により、CASではレアリティが高いブロックを待つピアが少なくなる。その結果、ピアがファイル復元をするのが容易となり全ピアがファイル復元を完了させるのに要する時間は短くなる。また、レアリティが高いブロックを待つピアが少ないためCASでは大半のピアがファイル復元に要する時間は大きく変わらない。シミュレーションでは全ピアが同時にブロック交換を始めるので、全ピアが同時にファイル復元するのが理想である。CASの方がファイル復元に要する時間差が短くピアによる格差が小さいため理想的な状況に近い。

途中参加ピアがいる場合は、本提案では3つの状態を想定してシミュレーションを行う。シードからファイルが共有されはじめてすぐでファイルの需要が最も高い初期状態、ファイルが共有されて時間が経ちピアの参加・離脱が安定した定常状態、多くのピアへファイルが広まった後の末期状態である。初期状態はファイル共有を希望するピアが最も多く、頻りにネットワークへピアが参加する。初期状態に参加したピアは図5のブロック交換開始から約300Roundに該当し、最初はネットワーク内にピアが存在しないため、ネットワークから離脱するピアが存在せず、全状態の中で最も参加ピア数が多い状態を想定している。定常状態はファイル共有の大半の時間を占めており、定常状態に参加したピアは図5の300~3,500Roundに該当し、ネットワークに参加するピア数と同等のピアがネットワークから離脱する状態を想定している。末期状態はファイル共有の希望者が少なく新規参加するピアが少ない。末期状態に参加したピアは図5の3,500Round以降に該当し、ネットワークに参加するピア数よりもネットワークから離脱するピア数の方が多く状態を想定している。

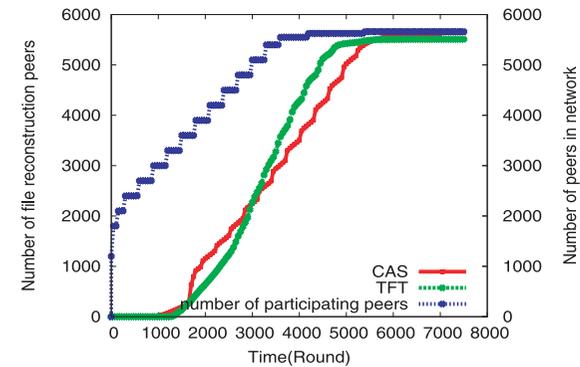


図 5 途中参加ピアがいる場合の平均ファイル復元時間

Fig. 5 Average file reconstruction time when peers join network on the way.

初期状態に参加したピアのファイル復元の様子は途中参加ピアがない場合と酷似している。途中参加ピアがない場合と同様の理由で、初期状態に参加したピアがファイル復元を完了させるのにかかる時間は短い。初期状態に参加したピアがネットワークから離脱する様子は1,000~2,000Roundのファイル復元率に表れている。本シミュレーションでは、途中参加ピアがない場合の評価からも分かるように、CASを使用するとブロックを収集し終えるまで約2,000Round要する。そのため、ピアがネットワークに参加してから離脱するまでに約2,000Roundの時間差が発生している。CASの方がTFTより、ファイル復元に要する時間が短いためファイル復元率が高い。結果、2,000Round時におけるファイル復元率がTFTは10%なのに対しCASは20%と約2倍のファイル復元率となった。

定常状態になるとTFTの方がCASよりもファイル復元に要する時間は短くなる。定常状態の間は、想定どおりの結果となりネットワークに参加するピア数と同等数のピアがネットワークから離脱し、CASのファイル復元率の傾きと途中参加ピア数の傾きが比例している。CASは途中参加ピアがない場合の考察で述べたように、同時に参加したピアどうしがファイル復元を完了させる時間に大きな差がない。そのため、同時に参加したピアどうしは近いタイミングでファイル復元に成功する。それに対しTFTは、定常状態の間はファイル復元率の傾きの方が途中参加ピア数の傾きよりも急である。TFTはピア離脱の制約を設けておらず、ピアはブロックが集まり次第ファイル復元を行い離脱する。そのため、定常状態においてはTFTの方がCASよりもファイル復元に要する時間が短くなるが、レアリティの高いブロックが短時間しか提供されなくなる問題が起き、末期状態でブロックのレアリ

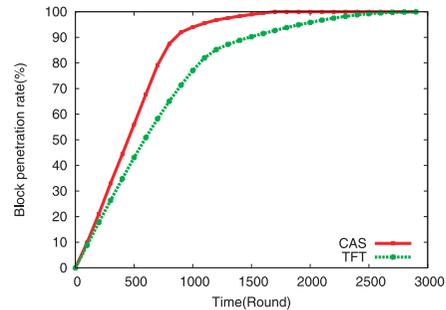


図 6 ブロックの平均普及時間の比較
Fig. 6 Compare average block penetration time.

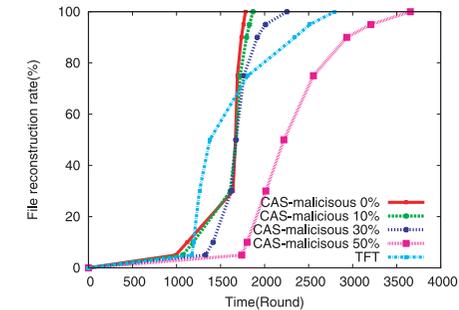


図 7 悪意あるピアが存在する場合の平均ファイル復元時間
Fig. 7 Average file reconstruction time when malicious peers exist.

リティ問題が考慮されていないために不都合が起こる。

末期状態では CAS の方が TFT よりファイル復元率が高い。末期状態においてピアがファイル復元を完了できるかはブロックのレアリティ問題を考慮したかで決まる。レアリティ問題を考慮しないとレアリティの高いブロックがネットワークから消失する。ブロックがネットワークから消失するとファイルを復元できない。本シミュレーションでは、7,500 Round 時におけるファイル復元率が CAS は 99.6%、TFT は 97.3%となった。

4.2.2 ブロックの平均普及時間

図 6 に CAS と TFT のブロックの平均普及時間の比較を示す。ブロックの平均普及率は CAS の方が TFT より高くブロックの普及速度が速い。これは、CAS では所持ブロック数の少ないピアがダウンロード帯域を有効活用しておりブロックの分散速度が速いためである。普及率 80~90%あたりで普及速度が低下するが CAS はブロックを効率的に分散させているため TFT より高いブロック復元率まで普及速度を保っている。本シミュレーションでは全ブロックが全ピアに普及するまでに要する時間を TFT より約 1,000 Round 短縮している。

4.2.3 悪意あるピアが存在する場合

図 7 に CAS において悪意あるピアが存在する場合の平均ファイル復元時間を示す。

図 7 より、悪意あるピアが多いほどファイル復元時間が長くなること分かる。また悪意あるピアの比率が大きくなるほどファイル復元に要する時間が長くなる。これは、悪意あるピアがリソースを提供しないためブロック交換の能率が下がるためである。しかし、本シミュレーションでは悪意あるピアが 30%存在する場合でも、全ピアがファイル復元に要す

る時間は CAS の方が悪意あるピアが存在しない場合の TFT より短い。これらから、悪意あるピアが存在する場合でも CAS の方が TFT よりもファイル復元時間が短く、ブロックのレアリティを考慮することはブロックを効率的に収集する手法として有用である。

5. おわりに

本論文では、ブロックを効率的に分散させる P2P ファイル共有手法 CAS を提案した。P2P ファイル共有では、ブロックがネットワークへ分散せず入手困難なブロックが発生することでファイル復元に要する時間が長くなるというブロックのレアリティ問題がある。そのため、ブロックのレアリティ問題を考慮しブロックを効率的に分散させる手法が必要である。

CAS では、TFT がブロック交換を行う 1 対 1 のピア間に制約を設けているのに対し、過去アップロード数情報を用いてブロック交換を行う 1 対多のピア間での制約を設けた。制約条件の必須アップロード数は各ピアの所持ブロック率に応じて動的に変化させた。これにより、ブロックのレアリティ問題の原因であるブロックの分散速度とピア離脱に対処できる。

シミュレーションによる評価の結果、途中参加ピアがない場合では、全ピアがファイル復元を完了するのにかかる時間が短縮するのを確認した。途中参加ピアがいる場合でも、ピア数が少ない末期状態でファイル復元を完了させるピア数が増えるのを確認した。CAS はブロックのレアリティ問題を考慮することで、効率的なブロック収集の障害要因であるレアリティの高いブロックの発生を抑制している。そのため、ファイル復元を完了できる確率が TFT よりも高い。ブロックの平均普及時間を比較した結果、CAS の方がブロックの分散速度が速くブロックのレアリティ問題を抑制しているのを確認した。悪意あるピアが存在す

る場合の平均ファイル復元時間を比較した結果、悪意あるピアがネットワーク全体の 30% 存在する場合でも、CAS の方が悪意あるピアが存在しない場合の TFT よりもファイル復元時間が短くなることを確認した。これらから、ブロックのレアリティ問題を考慮したブロックを効率的に分散させる P2P ファイル共有手法 CAS は、ブロックを効率的に収集する手法として有用である。

今後の課題として、ブロックのレアリティ問題における最適手法や、より効率的な P2P ファイル共有手法の検討を考えている。CAS はブロックのレアリティ問題の完全解決という観点において、まだ最適ではない。多くのファイル復元完了ピアによる同時離脱への対処、ブロックのレアリティによる分散優先度の設定、各ピアの通信回線の考慮などに改善の余地がある。また、今回考慮していないピア隣人の最適配置など、ブロックのレアリティ問題以外のアプローチからもファイル復元完了時間をさらに短縮できると考えている。

謝辞 本研究の一部はグローバル COE プログラム「アクセス空間支援基盤技術の高度国際連携」により行われました。

参 考 文 献

- 1) Ramaswamy, L. and Liu, L.: Free riding: A new challenge to peer-to-peer file sharing systems, *IEEE HICSS'03* (2003).
- 2) Cohen, B.: Incentives build robustness in BitTorrent, *Proc. 1st Workshop on Economics of Peer-to-Peer Systems* (2003).
- 3) Legout A., Urvoy-Keller, G. and Michiardi, P.: Rarest first and choke algorithms are enough, *Proc. 6th ACM SIGCOMM Conference on Internet Measurement*, pp.203–216 (2006).
- 4) Neglia, G., Presti, G.L., Zhang, H. and Towsley, D.: A Network Formation Game Approach to Study BitTorrent Tit-for-Tat, *NET-COOP 2007*, pp.13–22 (2007).
- 5) Rai, V., Sivasubramanian, S, Bhulai, S., Garbacki, P. and van Steen, M.: A multiphased approach for modeling and analysis of the BitTorrent protocol, *IEEE ICDCS'07* (2007).
- 6) Jun, S. and Ahamad, M.: Incentives in BitTorrent induce free riding, *Proc. 2005 ACM SIGCOMM Workshop on Economics of Peer-to-Peer Systems*, pp.116–121 (2005).
- 7) Locher, T., Moor, P., Schmid, S. and Wattenhofer, R.: Free riding in BitTorrent is cheap, *HotNets'06* (2006).
- 8) Levin, D., LaCurts, K., Spring, N. and Bhattacharjee, B.: Bittorrent is an auction: Analyzing and improving bittorrent's incentives, *ACM SIGCOMM'08*, pp.243–254 (2008).

9) 総務省：平成 21 年版 情報通信白書，ぎょうせい (2009).

10) Gkantsidis, C. and Rodriguez, P.R.: Network Coding for Large Scale Content Distribution, *IEEE INFOCOM*, pp.2257–2267 (2005).

(平成 21 年 9 月 2 日受付)

(平成 22 年 3 月 5 日採録)

推 薦 文

本論文は、P2P ファイル共有においてパーツ収集効率を上げるためのパーツのレアリティを考慮した P2P ファイル共有手法 CAS (Carrot and Stick) を提案している。提案手法はレアリティ問題の原因であるパーツの初期分散速度とピア離脱の 2 点に対処するものであり、シミュレーションによりその有効性が示されている。本論文の有効性は明白であり、推薦に値する。
(マルチメディア通信と分散処理研究会主査 串田高幸)



松本 敬 (学生会員)

2009 年慶應義塾大学理工学部情報工学科卒業。現在、同大学大学院理工学研究科博士前期課程在学中。P2P ネットワークの研究に従事。



遠藤 伶 (学生会員)

2008 年慶應義塾大学理工学部情報工学科卒業。2010 年同大学大学院理工学研究科博士前期課程修了。現在、同大学院博士後期課程在学中。P2P ネットワークの研究に従事。



重野 寛 (正会員)

1990年慶應義塾大学工学部計測工学科卒業。1997年同大学大学院理工学研究科博士課程修了。1998年同大学理工学部情報工学科助手(有期)。2003年同大学理工学部情報工学科助教授。現在、同大学理工学部准教授。博士(工学)。情報処理学会学会誌編集委員, 同論文誌編集委員, 同マルチメディアと分散処理研究会幹事, 同モバイルコンピューティングとワイアレス通信研究会運営委員等を歴任。ネットワーク・プロトコル, モバイルコンピューティング, ITS, ネットワーク・セキュリティ等の研究に従事。著書『コンピュータネットワーク』(オーム社), 『ユビキタスコンピューティング』(オーム社)等。電子情報通信学会, IEEE, ACM 各会員。