

## パーツのレアリティを考慮した P2P ファイル共有手法 CAS の提案

松本 敬<sup>‡</sup> 遠藤 伶<sup>†</sup> 高木 健士<sup>†</sup> 重野 寛<sup>‡</sup>  
慶應義塾大学理工学部<sup>‡</sup> 慶應義塾大学大学院理工学研究科<sup>†</sup>

P2P ファイル共有において、ピアのパーツ収集効率を上げる手法についての研究がされている。パーツとは、共有するファイルを予め決められたサイズに分割した断片のことである。そして、P2P ファイル共有ではレアリティが高く入手の難しいパーツが発生し、パーツ収集効率が下がるパーツのレアリティ問題がある。そこで、本稿では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 の提案を行う。CAS で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 点に対処することで、パーツ収集効率を上げる。さらに、シミュレーション評価を行い CAS のパーツ収集効率における有用性を示す。

### CAS:P2P file share method considering rarities of parts

Kei MATSUMOTO<sup>‡</sup> Rei ENDO<sup>†</sup> Kenji TAKAGI<sup>†</sup> Hiroshi SHIGENO<sup>‡</sup>  
Faculty of Science and Technology, Keio University<sup>†</sup>  
Graduate School of Science and Technology, Keio University<sup>‡</sup>

The method for improving the parts collection efficiency of peer in P2P file sharing is investigated. Parts are the divided fragments of shared files as which size is decided previously. And, there is a rarity problem of parts in P2P file sharing that degenerates the parts collection efficiency of peer by the parts that have high rarity and thus difficult to obtain it. The aim of proposal is to improve the parts collection efficiency of peer in P2P. We propose CAS that considers the rarities of parts. In the proposal, CAS deals both the initial dispersion speed and the peer departure to improve the parts collection efficiency. In addition, We evaluated the proposal by simulation and showed the availability of CAS for parts collection efficiency of peer.

## 1 はじめに

P2P 技術を用いた代表的なサービスとしてファイル共有が挙げられる。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（パーツ）が交換される。ピア間で交換を行い全てのパーツを集めて元のファイルを復元させることでファイルを共有している。P2P ファイル共有ネットワークに参加している各ピアはアップロード、すなわち他ピアへのパーツの送信とダウンロード、すなわち他ピアからパーツ受信を行う。また P2P 技術を使っているため、リソース（ファイルのアップロード帯域）をネットワークに参加しているピアが提供している。これにより、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も特長である。

そしてこれらのファイル共有サービスで問題となっているのが、パーツ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 問題である [1]。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ため、このようなピアが存在するとファイル共有の性能が落ちてしまう。代表的な P2P ファイル共有ソフトである BitTorrent [2] では、free-riding 問題の対策として TFT (Tit-for-Tat) [3, 4, 5] を用いている。

TFT では、予め決められた  $n$  個（一般的には 2）までのパーツのダウンロードは自由にできるが、それ以上のパーツ数をダウンロードしたい場合にはそのパーツ数だけアップロードを行わないと新たなパーツを入手できないような戦略となっている。この戦略によりピアはアップロードを行う必要が生じ free-riding 問題へ対応できる。

しかし、ファイル共有の抱える別の問題としてパーツのレアリティ問題が存在する。初期分散の速度、すなわちパーツを最初に配信する際にネットワークへの広がりが遅いと、パーツがネットワークに行き渡り難くなる。また、ネットワークからピアが離脱する際にそのピアが所持していたパーツが消失してしまうため、そのパーツのレアリティが高くなる。その結果パーツの種類によってネットワークに存在する数がばらつき、数の少ない入手の難しいパーツが発生してしまう。このようにネットワーク内に数が少ないパーツのことを本稿ではレアリティが高いパーツと呼ぶ。この問題によりピアが未所持のパーツを入手できない状況が起こり、全てのパーツを揃えるのに時間がかかってしまう。既存手法ではレアリティ問題を考慮していないためにネットワーク全体としてのパーツ収集効率が悪くなる。

そこで、本稿ではパーツのレアリティ問題を考慮した P2P ファイル共有手法 CAS (Carrot and Stick) を提案する。CAS では、各ピアの入手パーツ数に応じて必須アップロード数を変化させる。必須アップロード数とは、新規のパーツ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条件のことを指す。パーツ所持率が少ない間は必須アップロード数を 0 にすることで、CAS では無駄になっていたダウンロード帯域を有効活用できパーツの初期分散速度が速くなる。ピアの離脱に対しても、ピア離脱前の必須アップロード数を多く設定しダウンロードを制限することにより対応する。その結果パーツがネットワーク内を循環し易くなりレアリティ問題が抑制される。CAS により、パーツのレアリティを考慮しパーツ収集効率の改善が可能となる。さらにシミュレーション評価を行って、CAS のパーツ収集効率における有用性を示す。

## 2 パーツのレアリティ問題

P2P ファイル共有では、ファイルを共有するためにピア間でパーツ交換を行っている。このパーツをピア間で交換し、全種類のパーツを集めたピアは元のファイルを復元し入手することができる。

パーツのレアリティ問題は、ピアがパーツ交換を行っていく内にネットワーク内に存在するパーツ数がばらつくことで起こる。ばらつきが起こる原因は様々だが、ばらついた結果ネットワーク内にレアリティの高いパーツが発生する。ピアがレアリティの高いパーツを入手するのは困難なので、レアリティの

高いパーツが発生するとパーツ収集効率が低下する。

以下で、レアリティ問題発生の主な原因であるパーツの初期分散速度とピアの離脱について説明する。

### 2.1 パーツの初期分散速度

パーツの初期分散速度とは、最初にパーツが配信される際のネットワークへ広がる速度を意味する。この速度が速い場合は、広範囲のネットワークへとパーツが広がる。その結果、多くのピアが何らかのパーツを所持することになりネットワーク全体でパーツ交換が行われるようになる。

それに対して、初期分散速度が遅い場合はネットワークへパーツが広がり難く、始めの間は何のパーツも所持していないピアが大半となる。その結果、一部のピア間でしかパーツ交換が行われずネットワーク内に存在するパーツ数は増え難くなり、レアリティの高いパーツが発生しやすくなる。

### 2.2 ピア離脱

P2P ファイル共有では、ネットワークからのピア離脱が発生する。ピア離脱が発生する主な要因は、ファイルを復元したピアによるものである。パーツ交換を行うことで全種類のパーツを所持したピアは、ファイルを復元することが可能となる。ファイルを復元したピアはこれ以上ネットワーク内にいる必要がないために、ネットワークから離脱する。ピアが離脱してしまうと、そのピアが所持していたパーツがネットワークから消失するために各パーツのレアリティが高くなる。

## 3 関連研究

代表的な P2P ファイル共有ソフトである BitTorrent の既存研究の中で、free-riding 問題とパーツのレアリティ問題に関連する研究として Tit-for-Tat と BitTyrant がある。

Tit-for-Tat の特徴は、パーツをアップロードしなければダウンロードできなくなることである。この特徴により、パーツのダウンロードだけを行いアップロードを行わない free-riding 問題に対処できる。

具体的には、各ピアがパーツ交換を行う際に以下の式に従って行動することで実現している [6]。

$$DOWN_A - UP_{A \rightarrow B} \leq n \quad (1)$$

(1) 式はピア A とピア B がパーツ交換を行う時にピア A 視点で見た場合の式である。1 式の  $DOWN_A$  はピア A がピア B からダウンロードしたパーツ数

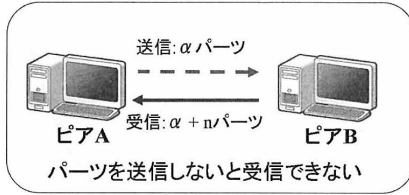


図 1: Tit-for-Tat

で、 $UP_{A \rightarrow B}$  はピア A がピア B へアップロードしたパーツ数である。この式に従いピア A と B は左辺が  $n$  以内ならばパーツ交換を続ける。つまり、 $n$  個までピア A はピア B から多くのパーツをダウンロードする。しかし、左辺が  $n$  個を超えたらパーツ交換は中止される。 $n$  個までのダウンロードを許容するのは、ある程度の許容幅がないとパーツを所持していない新規参加ピアがパーツをダウンロードできないからである。

$n$  個より多くのパーツをダウンロードしたい場合は、相手へパーツを送信し続ける必要がある。そのため他ピアへとパーツをアップロードすることの動機付けが行われており、free-riding 問題への対策となっている。図 1 にピア A とピア B がパーツ交換している様子を示す。

しかし、この手法ではダウンロード可能数がアップロード数に依存しているためにダウンロード帯域がアップロード帯域に制限されている。そのため、Tit-for-Tat 適用前よりもピアのダウンロードできるパーツ数が減る。本提案の目標であるパーツ収集効率の観点からこの手法では効率が悪い。

BitTyrant[7] では、最後の 1 ピースを十分に見つけることができるように BitTorrent を改良している。この研究の特徴は、ネットワークに参加している各々のピアが利己的な動きをすることである。アップロード速度の速い（貰えるパーツ数の多い）ピアと接続することを行動目標とし、自分にとって都合の良いピアへの接続を選択的に行う自分の利益を最優先させるシステムとなっている。そのため、このシステムは少数のユーザが使用する分には良いが、多数のユーザが使用するとシステム全体の性能を低下させてしまう可能性があることが分かっている。

具体的には、自分の提供するアップロード帯域を、入札ピア、すなわちアップロードを希望する他ピアに対してオークションにかけることで効率の良い接続を実現する。入札ピアは、自らがそのピアに対し

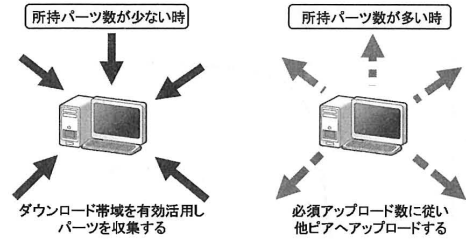


図 2: CAS

て提供できる帯域を入札する。そして、最終的に入札帯域量の多かった入札ピアから順に落札ピアとし提供帯域を分割して提供する。

## 4 提案

本稿では、P2P ファイル共有においてパーツ収集効率が悪くなるパーツのレアリティ問題に着目し、パーツのレアリティを考慮した P2P ファイル配信手法 Carrot and Stick (CAS) を提案する。CAS で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 点に対処することで、パーツ収集効率を改善する。

図 2 に CAS におけるパーツ交換の様子を示す。CAS では所持パーツ数の少ないピアは、ダウンロード帯域を有効活用しパーツの収集を行う。そして所持パーツ数が多いピアは、必須アップロード数に達するまでパーツのアップロードを行う。

### 4.1 ダウンロード帯域の有効活用

P2P ファイル共有では初期分散速度の速さがパーツのレアリティ問題の原因となっている。既存手法の TFT(Tit-for-Tat) では、ダウンロード帯域が制限されてしまうためにパーツの収集効率が悪くなっている。そこで、初期分散速度を速くすることでパーツのレアリティ問題を抑制する。その方法として、ピアのダウンロード帯域を有効活用する。

CAS では、ネットワークに参加したばかりで所持パーツ数が少ない場合には、パーツの初期分散速度を速くするために必須アップロード数を 0 とし自由にパーツのダウンロードを許す。

図 3 に CAS と Tit-for-Tat の時間と所持パーツ数の関係を示す。この図を見ると、ダウンロード帯域が制限されてしまっている Tit-for-Tat に比べ CAS では早い時間で多くのパーツを所持できるようになっている。その結果、ネットワーク内のパーツ数が増えレアリティが高いパーツの発生を抑制できる。

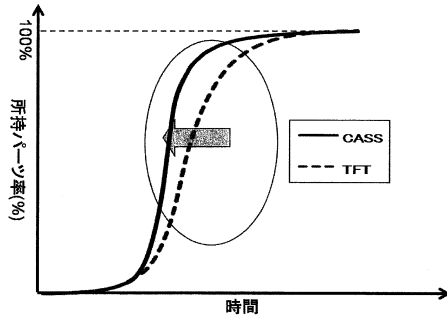


図 3: 初期分散速度の改善

## 4.2 ピア離脱の制約

ピア離脱が発生するとパーツのレアリティが上がりファイル収集効率が悪くなる。そこで、ピア離脱寸前に必須アップロード数を多く設定することにより制約する。具体的には、所持パーツ数が多いピアに対して必須アップロード数を多く設定する。

## 4.3 必須アップロード数の決定

CAS では必須アップロード数の値を変動させることによりピアの状況を制御している。ネットワークへ参加したばかりのピアに対しては必須アップロード数を 0 とし、所持パーツ数が多く離脱寸前のピアに対しては必須アップロード数を極力大きくするのが理想的である。

そこで、CAS では必須アップロード数の値を決めるために、値が急激に上昇する特性を持つ指数関数を用いることとした。しかし、指数関数をそのまま使うと、ネットワークに参加したばかりのピアにも小さい値ながら必須アップロード数が発生するため以下のような式に変形した。

$$\left(\frac{N^{100}}{2} - N^x\right) / N^{100} \times \text{全パーツ数} \times 2 \quad (2)$$

$N$  とは指数関数の底を、 $x$  は所持パーツ率をそれぞれ表している。この式を用いることで、特定の所持パーツ率 (図 4 の  $g$ ) になるまではアップロード数は 0 となり、ダウンロード帯域を有効活用できる。また、 $N$  の値を変更させることにより指数関数の傾きが変わるために、 $N$  の値により必須アップロードを 0 から切り替えるタイミングを調整できる。

図 4 に CAS における所持パーツ数と必須アップロード数についての関係を示す。横軸はファイル復

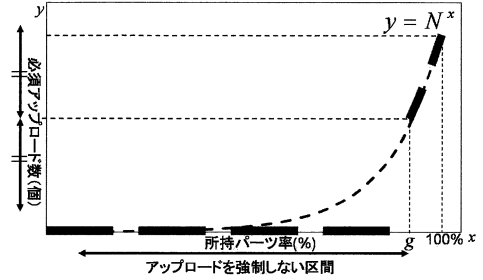


図 4: CAS の特性

元に必要なパーツ数と所持パーツ数との比率を表す所持パーツ率、縦軸は必須アップロード数である。

## 4.4 動作アルゴリズム詳細

ここでは未所持パーツを探しているピアを *Receiver* ピアとし、そのパーツを所持しているピアを *Sender* ピアとする。

### 1. パーツ交換の開始

パーツ交換は、*Receiver* ピアの未所持パーツを持っている隣人ピア *Sender* に対して、パーツ要求と *Receiver* ピアの所持パーツ数、過去アップロード合計数情報を送信することで開始する。過去アップロード合計数とは、今までに *Receiver* ピアが他ピアに対してアップロードを行ったパーツの合計数である。

### 2. 必須アップロード数の計算

パーツ要求を受け取った *Sender* ピアは、*Receiver* ピアの所持パーツ数情報から *Receiver* ピアの現在の必須アップロード数を式 2 を使い計算する。

### 3. アップロード数の比較

*Sender* ピアは、*Receiver* ピアの必須アップロード数と過去アップロード合計数を比較する。

その結果、必須アップロード数よりも過去アップロード合計数が多かった場合は、*Receiver* ピアは新規パーツを入手出来る状態であると判断する。そして、*Sender* ピアは *Receiver* ピアに対して要求パーツのアップロードを行う。

しかし、過去アップロード合計数が必須アップロード数よりも少ない場合は、*Receiver* ピアは新規パーツを入手できる状態ではないと判断し要求パーツのアップロードを行わない。

## 5 シミュレーション評価

CAS のパーツ収集効率の有用性を示すために、既存手法 Tit-for-Tat との比較評価を行った。

### 5.1 シミュレーション条件

表 1 にシミュレーションで使用了条件を示す。ダウンロード数やアップロード数は、通常ピアは ADSL、高性能ピアは FTTH から性能比を算出した [8]。

また、本シミュレーションは単位時間を Round とする。各ピアは Round 単位でパーツのアップロードとダウンロードを行っている。ピアの能力は、各ピアが Round 辺りにどのくらいのパーツを他ピアへアップロードできるかを示している。

評価項目として、ファイル分割数を変化させた際のピアの平均ネットワーク離脱率と離脱時間をとった。シミュレーション回数は各 10 回行った。

表 1: シミュレーション条件

総ピア数	1000
ファイル分割数	1000 ~ 10000 パーツ
隣人更新周期	3Round
最大隣人数	10
高性能ピア最大アップロード数	15 パーツ
高性能ピア最大ダウンロード数	15 パーツ
通常ピア最大アップロード数	3 パーツ
通常ピア最大ダウンロード数	10 パーツ
通常ピアと高性能ピアの比率	1:2

#### 5.1.1 シミュレーションシナリオ

- ネットワーク内のパーツを収集しファイルを復元することを目的としたピアを 1000 参加させる。
- ファイルを 1000~10000 パーツに分割し、毎 Round サーバから 10 ピアに配信する。
- 各ピアが隣人リスト内のどの隣人へパーツを渡すかはランダムに決定する。
- 隣人へのアップロードは Round 単位で行い、既存方式と CAS との違いを以下にまとめた。

##### — 既存方式

パーツ交換相手から以前に貰ったパーツ数を参考に、既存の式に従いアップロード上

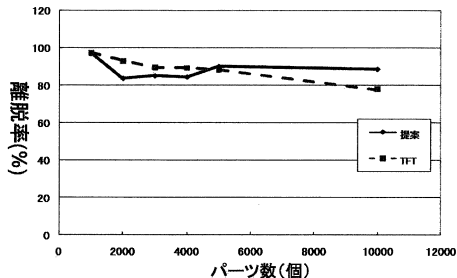


図 5: パーツ数とピア離脱率の関係

限数まで配信する。

##### — CAS

アップロードする相手をランダムに決定し、アップロード上限数までランダムにパーツを配信する。

- 隣人からのダウンロードは Round 単位で行い、既存方式と CAS との違いを以下にまとめた。

##### — 既存方式

パーツ交換相手のアップロード分だけ、自らのダウンロード上限数まで行う。

##### — CAS

2 式に従い、ダウンロード可能数分だけダウンロード上限数まで行う。

- 全種類のパーツを所持するピアはネットワークから離脱する。
- 離脱ピア数の変動が 300Round 無くなった時点でシミュレーションを終了する。

## 5.2 シミュレーション結果

ファイル分割数の変化によるピアの平均ネットワーク離脱率と離脱時間から、パーツ数の変化がシステムに与える影響について考察する。

### 5.2.1 離脱率

図 5 にパーツ数を変化させた場合のピア離脱率について示す。図において横軸はパーツ数、縦軸はピアの平均ネットワーク離脱率を示している。

ネットワーク離脱率というのは、パーツを全種類所持することでファイルを復元しネットワークから

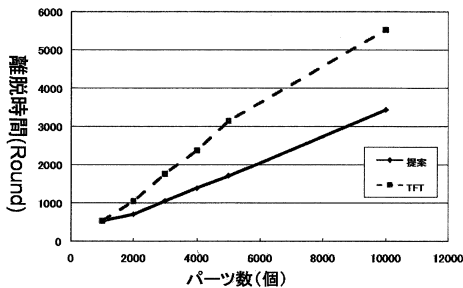


図 6: パーツ数とピア離脱時間の関係

離脱したピア数を表しており、具体的には離脱したピア数を参加ピア数で割った値である。

既存手法ではパーツ数が多くなるに従い、ピアの平均ネットワーク離脱率も低下している。それに対し、CASではパーツ数が多くなることによる影響はあまり見られない。P2P ファイル共有では、パーツの種類が多くなるほどレアリティの高いパーツが発生し易くなる。そのため、既存手法ではパーツ数が多くなるにつれ全種類のパーツを所持できなかったピアが増加し、離脱率の低下が起こっている。しかし、CASではレアリティの高いパーツが発生し難くなっているためにパーツ数増加による離脱率の影響をあまり受けていない。

### 5.2.2 離脱時間

図 6 にパーツ数を変化させた場合のピア離脱時間について示す。

ネットワーク離脱時間は、パーツを全種類所持することでファイルを復元しネットワークから離脱するまでの時間である。

既存手法も CAS もパーツ数が増えることにより離脱時間が増えるのは同じである。しかし、CASでは既存手法に比べて離脱時間が約 60% 早くなっているのが分かる。CAS はパーツのレアリティ問題を考慮しているのでレアリティの高いパーツが発生し難くなっている。そのため、既存手法より未所持パーツを入手し易くなっており早くパーツを収集することができる。

## 6 おわりに

本稿では、P2P ファイル共有におけるレアリティを考慮した free-riding 対策手法 CAS を提案した。P2P ファイル共有ではレアリティの高いパーツが発生することでパーツ収集効率を下げるパーツのレア

リティ問題が存在する。そのため、レアリティを考慮しパーツの収集効率を改善する手法が必要である。

CAS では、レアリティ問題の原因となっていた初期分散速度の問題を改善するためにダウンロード帯域を有効活用できるようにした。そして、ピア離脱の問題を改善するために必須アップロード数を設定した。これにより、過去アップロード合計数が必須アップロード数を上回らない限り新規パーツをダウンロードできなくなる。

CAS をシミュレーションにより評価した結果、パーツ数を増やしてもファイルが復元できなくなるピア数に影響は見られず、離脱時間に関しては既存手法に比べ約 60% の速度改善を確認した。

以上より CAS は P2P ファイル共有において効率的なパーツ収集手法として有用である。

### 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] L.Ratnasamy and L. Liu. Free Riding: A new challenge to peer-to-peer file sharing systems. HICSS, 2003.
- [2] Bram Cohen. Incentives build robustness in bit-torrent. P2P Economics Workshop, Berkeley, CA, 2003.
- [3] A.Legout, G.Urvoy-Keller, and P.Michiardi. Rarest first and choke algorithms are enough. IMC, 2006.
- [4] G.Neglia, G.L.Presti, H.Zhang and D.Towsley. A network formation game approach to study BitTorrent tit-for-tat. NET-COOP, 2007.
- [5] V.rai, S.Sivasubramanian, S.Bhulai, P.Garbacki and M.van Steen. A multi-phased approach for modeling and analysis of the BitTorrent protocol. ICDCS, 2007.
- [6] S.Jun and M.Ahamad. Incentives in BitTorrent induce free riding. P2PEcon, 2005.
- [7] M. Piatek, T. Isdal, T. Anderson, A.Krishnamurthy and A. Venkataramani. Do incentives build robustness in BitTorrent?. NSDI, 2007.
- [8] 大橋省吾, 大根田進 and 町田達彦. FTTH 時代を担うファースト 1 マイルソリューション. SWCC グループ, 2005.