Vol.2014-DPS-161 No.4
Vol.2014-EIP-65 No.4
2014/9/18

BitTorrent型配信システムにおけるRateless Codingを用いた希少ピース拡散手法についての一検討

藤本 章宏[1,a)]　廣田 悠介[2,b)]　戸出 英樹[3,c)]

概要：高いスケーラビリティをもつ大規模コンテンツ配信基盤として，ユーザ端末間でデータを交換するP2P型配信システムが広く利用されるようになっている．P2P型配信システムでは，ユーザ端末は自由にシステムへの参加・離脱が可能な一方，それにより特定のデータの入手可能性が低下する問題が指摘されている．そこで，本稿では，離脱耐性向上を目的とし，Rateless Codingと呼ばれる冗長符号化手法を用いたデータ拡散手法を提案する．特に，符号化効率と完全なピースの入手確率のバランスを考慮し，希少性に基づき選択された複数のピースに対し，当該データの入手確率が向上するように符号化を行う．また，提案手法の有効性を，計算機シミュレーションにより評価する．

キーワード：P2P, 希少ピース, レートレス コーディング, 離脱耐性

A Study on Diffusion Method of Rare Pieces using Rateless Coding on BitTorrent-like Distribution System

Akihiro Fujimoto[1,a)]　Yusuke Hirota[2,b)]　Hideki Tode[3,c)]

Abstract: P2P content distribution systems become popular for large scale content distribution because of their high scalability. In P2P content distribution systems, users can join and leave the distribution networks whenever they want. However, such churn could decrease the availabilities of some pieces of the content. To improve the torelance to node leave, this paper proposes a content diffusion method using rateless coding, which is a variation of error correction codes. The proposed method encodes selected pieces of content based on their rarity such that the rare pieces can be available with high probability. This rarity-based encoded piece selection improves the encode efficiency and the piece availability. The effectiveness of the proposed method is evaluated through computer simulations.

Keywords: P2P, Rare Piece, Rateless Coding, Node Leave Tolerance

1. はじめに

近年，ネットワークの高速化に伴い，多数のユーザがネットワークを通し，多様なコンテンツにアクセスできるようになった．このような背景の下，大規模なコンテンツ配信を可能とする配信基盤として，高いスケーラビリティをもつP2P型配信システムが広く利用されるようになっている．

P2P型配信システムでは，ユーザ端末（ピア）同士が，サーバ等を介さず直接データを交換することでコンテンツを拡散する．クライアント・サーバ型のシステムと異なり，特定の端末に配信負荷が集中することがないため，高いスケーラビリティをもつ配信ネットワークを低コストで構築

[1] 和歌山大学システム情報学センター
Center for Information Science, Wakayama University
[2] 大阪大学大学院情報科学研究科
Graduate School of Information Science and Technology, Osaka University
[3] 大阪府立大学大学院工学研究科
Graduate School of Engeineering, Osaka Prefecture University
[a)] fujimoto@center.wakayama-u.ac.jp
[b)] hirota.yusuke@osaka-u.ac.jp
[c)] tode@cs.osakafu-u.ac.jp

できるという特徴がある．

コンテンツをダウンロードしたいユーザは，まず，そのコンテンツを配信している P2P ネットワークに参加する．そして，隣接ピアとピースと呼ばれるコンテンツの断片を交換し，当該コンテンツの全てのピースを揃えることができれば，ダウンロード完了となる．ダウンロードを完了したピアは，そのまま P2P ネットワークに残ってコンテンツの拡散に協力しても良いし，ネットワークを離脱しても良い．

P2P 型配信システムは，ピア間でのコンテンツ交換により高いスケーラビリティを達成できる一方で，ユーザの参加・離脱は自由であるため，通常のサーバを介した配信と比較し，配信能力が保証されないという弱みがある．例えば，ある一つのピアしか保持していないピースが存在する場合，そのピアが P2P ネットワークを離脱してしまうと，他のピアは全てのピースを揃えることができなくなってしまう．

この課題に対して，代表的な P2P 型配信システムである BitTorrent [1], [2] は，Rarest-First と呼ばれるピース交換方式をとっている．本方式では，各ピアは，ピースを揃える毎に，隣接ピアに対してそれを広告する．ピアは，この広告を元に隣接ピアが保有しているピースを管理し，次にダウンロードすべきピースとして，自身の隣接ピアの中で，保持しているピアが最も少ないピース，すなわち，最も拡散されていないピースを選択する．各ピースが均等に拡散されることで，特定のピアの離脱により，ピースが揃わなくなる可能性を低減することができる．

しかしながら，多数のピアが一斉に離脱する環境においては，Rarest-First 方式を用いたとしても，コンテンツの全ピースを入手できる可能性が低下してしまう．例えば，文献 [3] は，上述の環境として，動画のチャンクを順にダウンロードしながら再生を行う動画配信サービスを挙げ，Rarest-First 方式では配信能力が低下することを示している．このような配信能力の低下は，Rarest-First 方式が局所的な P2P ネットワーク内のピース保有状況に基づいていることによる．このため，局所的には均等にピースを拡散しているにも関わらず，P2P ネットワーク全体からみるとピースの拡散状況に偏りが発生し，ピアの大規模な離脱により一部のピースが完全に失われることが起こり得る．P2P 型配信システムを，より高品質な大規模コンテンツ配信基盤として確立するためには，ピアの離脱への耐性を高める必要がある．

本稿では，上述の課題の解決策として，Rateless Coding [4] と呼ばれる冗長符号化手法を用いたデータ拡散手法を提案する．Rateless Coding とは，定数個のデータブロックから，任意の数の符号化ブロックを生成可能な冗長符号化方式である．受信側では，生成された符号化ブロックの内，一定数以上の任意のブロックを入手できれば，元のデータブロック群を復元することができる．復元能力が入手したブロックの数にのみ依存するため，特定のピアの離脱が配信ネットワークに与える影響が低減される．特に，提案方式は，符号化効率と完全なピースの入手確率のバランスを考慮し，希少性に基づき選択された複数のピースに対して符号化を行う．さらに，特に優先的に拡散すべきピースを，重点的に符号化ブロックに含める．これにより，幅広い環境における，P2P 型配信システムの離脱耐性の向上を図る．

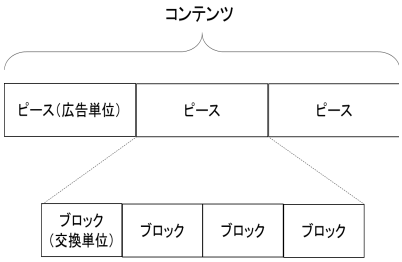

図 1 BitTorrent におけるコンテンツ分割

2. 関連研究

2.1 BitTorrent

BitTorrent [1], [2] は P2P 型コンテンツ配信システムの代表格であり，Sandvine 社の 2014 年上半期のレポート [5] によれば，アジア太平洋地域の固定網における BitTorrent トラヒックのシェアは約 27%に上ると報告されている．

BitTorrent では，同一コンテンツを要求するユーザ同士で配信ネットワークを構築する．このネットワークを構築するピアのうち，完全なコンテンツを保持しており，他のピアへのアップロードのみを行うピアを seeder，コンテンツを要求しており，他のピアとピースの交換を行うピアを leacher と呼ぶ．ネットワーク内のピアは tracker に管理されている．新しく配信ネットワークに参加したいピアは，tracker から他ピアの IP アドレスを受け取り，その情報を基にコネクションを張ることで，ネットワークに参加する．

ネットワークに参加した後，どのピアとコンテンツを交換するかを決定する必要があるが，BitTorrent では，Tit-For-Tat と呼ばれる戦略が取られている．この方式では，自身に多くのブロックをアップロードしてくれたピアに対し，返礼として多くのブロックをアップロードする．これにより，アップロードを行わないピアは淘汰され，より多くのピアがアップロードに貢献するようになるため，ネットワーク全体の配信効率を向上させることができる．

BitTorrent は，効率的にコンテンツを配信するために，コンテンツを細かく分割して管理する (図 1)．まず，コンテンツはピースと呼ばれる単位に分割され，各ピアはピース単位で自身の受信状況を広告する．ピアは，他ピアのピースの保有状況を基に，次にダウンロードするピースを決めることで，複数のピアからの並行ダウンロードを簡易

化できる．また，各ピースはさらにブロックと呼ばれる単位に分割される．ピア同士がサイズの小さなブロック単位で交換することで，帯域を有効に利用することができる．

安定した配信ネットワークを実現するためには，受信すべきピースの選択手法が重要となる．BitTorrent では，なるべく各ピースが均等に拡散されるよう，希少なピースを優先的にダウンロードする，Rarest-First と呼ばれる方式がとられている．前述の通り，各ピアは，自身のピース保有状況を隣接ピアに広告している．これにより，ピアは隣接ピアのピース保有状況が把握できるため，保有しているピアが最も少ないピースを算出できる．上記手法により，各ピアが自身の周囲で希少なピースを優先してダウンロードすることで，特定のピアの離脱が，ネットワーク全体に与える影響を低減する．

2.2 符号化を利用したコンテンツ交換手法

スループットの向上や離脱耐性の向上を目的として，データの符号化を P2P 型配信システムに適用する試みも行われている [3], [6], [7], [8]．このような取り組みの代表例として，Random Linear Network Coding [9] を利用した P2P 型配信システムである Avalanche [6] が挙げられる．Avalanche では，ピアは受信済みの符号化ブロック群に対して，ランダムに選択した係数を用いた線形結合による符号化を行い，隣接ピアに送信する．各ピアは，十分な符号化ブロックを受信できれば，元のデータを復元することができる．文献 [6] では，シミュレーションにより，Avalanche が，Network Coding を施さない場合と比較し，2-3 倍のスループットを獲得できることを示している．文献 [7] では，Rateless Coding を，P2P 型のライブストリーミングシステムに適用している．この方式では，各ピアはピースを復号した後，独自にそのピースを再符号化して隣接ピアに転送する．これにより，複数のピアから同一の情報を受信する確率を低減できると共に，離脱耐性を向上させている．

一方，文献 [3], [8] では，Rarest-First 方式を意識した Rateless Coding, Network Coding をそれぞれ提案している．これらの方式は，BitTorrent と同様に，コンテンツをピースに分割し，ピースをブロックに分割する．文献 [3] では，各ピアは，BitTorrent と同様の手法で希少なピースを要求する．要求先のピアは当該ピースに Rateless Coding を施し，符号化ブロックとして送信する．これに加え，隣接ピアが完全なピースを保有していない場合は，要求元ピアはその隣接ピアが当該ピースの符号化ブロックを保有していないか確認し，保有していればその符号化ブロックを転送するよう要求することで帯域の有効利用を実現している．文献 [8] では，各ピアは，隣接ピアの，ピース毎の保有ブロック数を把握しており．最もブロックの総数が少ないピースを選択して，そのピースを要求する．この際，ブロック数は通常のデータブロックと符号化ブロックを合わせて計算する．送信要求を受けたピアは，自身が保有する当該ピースのブロック群に対して Random Linear Network Coding を施し，符号化ブロックとして送信する．

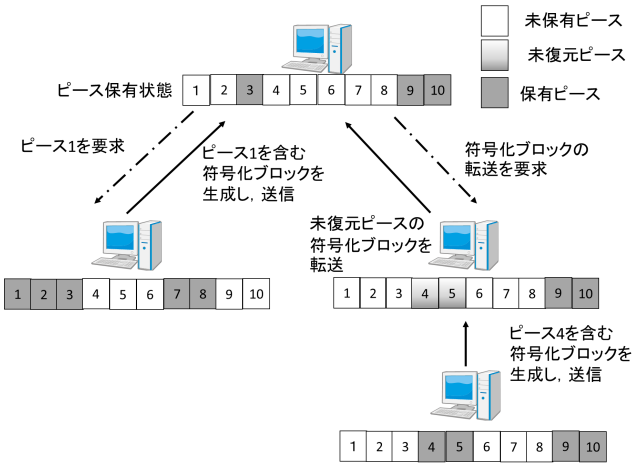

図 2 提案方式の概要

本稿で提案する手法は，文献 [3], [8] の手法と，ピースの希少性を考慮するという概念を共有している．しかしながら，これらの手法がピース内のブロックに対して符号化を行うのに対して，提案手法は，複数のピースを符号化対象としながら，希少なピースを優先的に符号化するという点で異なる．これにより，提案方式では，符号化効率と完全なピースの入手確率をバランス良く改善し，効率的に符号化ブロックを拡散することで，P2P 型配信システムの離脱耐性を向上できる．

3. Rateless Coding を用いた希少ピース拡散手法

3.1 概要

本稿では，ピアの離脱による配信能力の低下を改善するために，Rateless Coding を用いたデータ拡散手法を提案する．提案方式では，送信されるピース群は，要求元と要求先の双方の Rarest-First アプローチを元に選択される．選択されたピース群は，個別のブロックとしてではなく，当該ピース群の符号化により生成された符号化ブロックとして送信される．

提案方式の概要を図 2 に示す．各ピアは，BitTorrent と同様に，隣接ピアの受信状況から希少なピースを算出し，そのピースを保持しているピアにそれを要求する（図 2 左側の通信）．ピースの要求を受け取ったピアは，要求された希少ピースと，自身が算出した希少なピースの双方に基づき，Rateless Coding による符号化を行う．符号化は，選択された希少ピース群からランダムに複数のブロックを選出し，それらの排他的論理和をとることで行われる．こうして得られたデータを，「符号化ブロック」と呼ぶ．要求先ピアは，要求毎に，逐次符号化ブロックを生成し，送信する．完全なピースを復元できた場合，希少なピースを再算

出し，次のピースの要求を行う．

一方，自身にとって有用なピースをもたないピアを発見した場合は，そのピアがまだ復号できていない符号化ブロックを転送するように要求する（図 2 右側の通信）．転送要求を受けとったピアは，自身がキャッシュしている符号化ブロックのうち，まだ要求元ピアに転送していないものをランダムに選択し，転送する．これにより，送信すべきピースを保有していないピアもピースの拡散に貢献できる．

Rateless Coding では，復元能力が入手したブロックの数にのみ依存するため，特定のピアがもつピースが復号のボトルネックとなる可能性が少なくなり，ピアの離脱が配信ネットワークに与える影響を低減できると期待できる．また，複数のピアから符号化ブロックを受信する場合において，ピア間での綿密な同期を行う必要が無いことも利点である．本稿では，Rateless Coding として，符号化・復号処理が簡易な，LT(Luby-Transform) 符号 [10] を用いる．

提案方式では，既存の Network Coding，及び Rateless Coding を利用した P2P 型コンテンツ配信方式 [3], [6], [7], [8] と異なり，2 つのピアのピースの希少性に基いて選択した複数のピースに対して，LT 符号を適用する．これには，以下の利点がある．

(1) LT 符号の効率化
LT 符号は，符号化対象のブロック数が少ない場合，元のブロックを復元するために必要な符号化ブロック数が多くなることが知られている [10]．一方で，ブロック数が過度に多い場合，復号処理が複雑になる．提案方式では，符号化対象となるブロック数が適切になるようにピースを選択することで，符号化処理のコストと，オーバーヘッドのバランスをとることができる．

(2) 符号化ブロックの重複確率の低減
LT 符号は，復号に十分な符号化ブロックが揃うまで，データの復元はほとんど行われないという特性がある．このため，コンテンツ全体に対して符号化を行うと，長時間ピースを復元できない状況が続き，その間は，受信した符号化ブロックを他ピアに転送することになる．しかしながら，複数のピアから符号化ブロックを転送してもらう場合，それらの間で重複が発生してしまう可能性が高くなる．提案方式では，符号化対象のピースを限定するため，符号化ブロックの重複を抑制できる．

(3) 希少ピースの選択範囲の拡大
提案方式では，送信側・受信側双方が自律分散的に算出したピースの希少性に基づき，符号化が行われる．このため，ピースの拡散状況に局所的な偏りがあった場合でも，その影響を軽減し，配信ネットワーク全体に対してより有用な符号化ブロックが構築される．

上記の特徴から，提案方式を用いることで，各ピースの情報の拡散速度を向上させ，P2P 型配信システムの離脱耐性を改善できると期待できる．以下では，提案方式に基づく符号化手法について詳述する．

3.2 ピースの希少性に基づく符号化候補の選出

ピースの希少性は，自身の隣接ピアのピース保有状況に基いて決定される．このピース保有状況は，ピア同士が新しく隣接関係になった際に交換され，その後はいずれかのピースを揃えるたびに隣接ピアに通知することで更新される．提案方式では，各ピアはピース毎に，自身と隣接ピアの中で当該ピースを保有しているピア数を算出し，この保有数について昇順にソートしたリストを管理する．

提案方式では，送信側・受信側双方が算出したピースの希少性に基づき，符号化候補を選出する．ピア A がピア B に対して新しいピースを要求する場合について考える．ピア A が要求する符号化候補ピース数を N_{A} とすると，ピア A は，自身が管理する希少ピースリストを先頭から辿り，ピア A 自身が保有しておらず，かつ，ピア B が保有するピースを N_{A} 個だけ順に選出する．ただし，保有ピア数が同じピース，つまり，希少性が同等のピースがある場合は，ランダムに選択するものとする．

次に，ピア A は，選出したピースをピア B に要求する．このとき，要求メッセージには，ピア A が選出したピースの識別子が，保有数について昇順，つまり，希少性が高いピースから順に格納されているものとする．ピア B は，同様に自身が管理する希少ピースリストから，ピア A が保有しておらず，かつ，ピア A に要求されたピースではない N_{B} 個のピースを符号化候補として選出する．以上によって得られた $N_{\mathrm{A}} + N_{\mathrm{B}}$ 個のピースを，最終的な符号化候補とする．

3.3 ピースの希少性を考慮した符号化手法

提案方式では，要求元ピアから要求があったピースを，要求先ピアが選択したピースよりも優先的に拡散すべきピースであるとみなす．この仮定を，符号化・復号処理に反映させる手法として，本方式では，特定のデータブロック群を重点的に符号化可能な，LT 符号ベースの Expanding Window Fountain Code(EWF 符号) [11] を利用する．

EWF 符号は，ウィンドウによって，符号化対象のブロック群を区切り，ウィンドウ内のブロックに対して Rateless Coding を適用する手法である．この手法では，複数のウィンドウと，それに対応した符号化器を用意し，符号化ブロック毎に異なるウィンドウを用いて符号化を行う．このとき，重点的に符号化したいブロック群が，より多くのウィンドウに含まれるようにすることで，当該ブロック群をより多くの符号化ブロックに含めることが可能となる．その結果，重点的に符号化するブロック群は，その他のブロック群と比較し，より高速に復元することができる．

提案方式における EWF 符号のイメージを図 3 に示す．

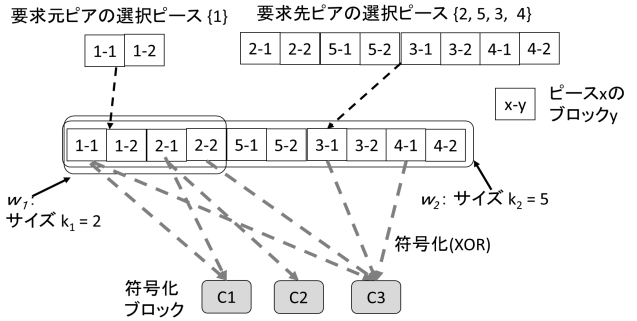

図 3 ピースの希少性を考慮した符号化ブロック生成

前述の通り，本稿では，符号化器として LT 符号化器を用いる．まず，ピア B は，ピア A, B によって選択されたピースを符号化バッファに送り，初期化する．このとき，より優先度の高い要求元であるピア A が選択したピース群が先頭に配置され，続いてピア B が選択したピース群が配置される．また，各ピアが選択したピース群の中では，保有ピア数の少ないピースほど先頭側に配置する．ピース毎のブロックは，ピースを分割した際の並びのまま配置される．ただし，提案方式では，ウィンドウはピース単位で設定されるため，ピース内のブロックの順序は符号化に影響しない．次に，優先度順にソートされたピースについて，ウィンドウ群 $W = \{w_1, w_2, \dots w_L\}$ を設定する．このとき，ウィンドウ $w_l, 1 \leq l \leq L$ のサイズを k_l とし，$1 < l \leq L$ において，$k_{l-1} < k_l$ を満たす．ウィンドウ w_l は符号化バッファの先頭から連続した k_l 個のピースによって構成される．つまり，ウィンドウは先頭から末尾に向かい，以前のウィンドウを内包しながら順次拡大される．これにより，重点的に符号化したい高優先ブロックが多くのウィンドウに含まれる編成が実現される．

初期化を終えると，符号化ブロックの生成を開始する．ピア B は，ウィンドウ群 W から，符号化に用いるウィンドウをランダムに選択する．その後，ピア B は選択したウィンドウ w_l に対応する LT 符号化器 $Encoder_l$ によって定められた確率分布に従って，符号化ブロックに含まれるブロック数 d を決定する．本稿では，従来の LT 符号に従い，$Encoder_l$ が用いる確率分布として，最大値 k_l，許容失敗率 δ_l, 符号化パラメータ c_l の Robust Soliton Distribution $\Omega(k_l, \delta_l, c_l)$ [10] を用いる．最後に，ウィンドウ w_l に含まれるピースのブロック群からランダムに d 個のブロックを選択し，それらの排他的論理和をとることで，符号化ブロックを生成する．

要求元であるピア A は，十分な数の符号化ブロックを受信すると，既に受信済みの符号化ブロックと復元済みのデータブロックを用いて，受信した符号化ブロックの復号を試みる．これによりいずれかのピースを復元することができた場合，自分がそのピースを入手したことを隣接ピアに広告する．

復号に十分な符号化ブロックを受信できていない場合は，ピア A は，同じピース群を引き続き要求する．要求先のピア B は，ピア A からの要求メッセージに含まれるピース識別子が直前のメッセージのものと変わっていない場合，初期化処理をスキップし，符号化ブロックを生成する．

3.4 符号化ブロック重複回避のための拡張方式

3.3 章で述べたように，参加直後のピアのように，いずれのピースも復元できていないピアは，専ら符号化ブロックの転送を行う．特に，コンテンツ配信を開始した直後は，ピースを揃えているピアがほとんど存在しないため，多くのピアが符号化ピースを転送した結果として，重複ブロックが多数発生する可能性が高い．

このような状態を回避するため，本稿では，拡張方式として，従来の Rarest-First 方式とのハイブリッド手法を提案する．本拡張提案では，保有ピース数の閾値 T_{rarest} を下回っている限り，符号化ブロックではなく，Rarest-First 方式に則ってブロックを要求する．

4. 性能評価

4.1 シミュレーション環境

以上の提案を，計算機シミュレーションによって評価した．本シミュレーションでは，初期 seeder は一つとし，ピアの到着率は $\lambda_0 \exp^{-\frac{t}{\tau}}$ [12] に従い，時間 t とともに減少するモデルを用いた．ただし，λ_0 は初期到着率，τ は減衰率であり，それぞれ 10，50 と設定した．また，ピアの接続数は，最大値を 8, 最小値を 4 とした．ピアは参加時に，接続数が最小値に達するまで，ランダムに選択した他のピアと接続を要求する．ただし，簡単化のため，Tit-For-Tat 戦略はとらず，全ての隣接関係にあるピアは，常に互いのピースを交換するものとした．また，隣接ピアが離脱した場合は，速やかにランダムに他のピアを選択し，接続するものとした．

コンテンツのサイズは 16MB とし，ピースのサイズを 256KB, ブロックのサイズを 16KB, ピース当たりのブロック数を 16 と設定した．各ピアの送信能力は等しく，一つの隣接ピアに対して，単位時間当たりに 3 ブロックを送信できるものとした．

提案方式のパラメータである，ウィンドウ数 L を 2 とし，各ウィンドウのサイズ k_l は外部から与えられるものとした．ウィンドウ w_1 が選択される確率 $P(w_1)$ のデフォルト値は 0.2 とした．符号化器 $Encoder_1, Encoder_2$ が用いる Robust Soliton Distribution Ω_1, Ω_2 は，それぞれ $\Omega(k_1, 0.5, 0.02), \Omega(k_2, 0.1, 0.02)$ とした．拡張方式の保有ピース数が閾値 T_{rarest} は，0.5 に設定した．また，本シミュレーションでは，要求元が選択する符号化候補ピース数を 1 に固定し，要求先が選択する符号化候補ピース数を $k_2 - 1$ とした．ピアの離脱に関しては，以下に示すシナリ

オを用いた．

- seeder, leecher に関わらず，全てのピアは率 μ で P2P ネットワークを離脱する．
- 上記に加え，開始から 60 秒後に，一定の割合 (Massive Leave Ratio) のピアが一斉に離脱する．

4.2 シミュレーション結果

以上の条件の下，本稿では，離脱耐性を測る指標として．コンテンツのダウンロードを完了したピアの割合 (ダウンロード完了率: Completion Ratio) について評価した．また，コンテンツのダウンロードを完了したピアの平均ダウンロード時間（Completion Time）についても評価した．ただし，ダウンロード完了率の算出には，一斉離脱したピアを含まないものとした．

図 4-7 に，Rarest-First 方式，及びコンテンツ全体に LT 符号化を施した場合 (LT) との比較結果を示す．ただし，提案方式を proposal, 3.4 章で提案した拡張方式を proposal2 で表しており，$k1, k2$ は符号化のウィンドウサイズを表している．

まず，離脱率 $\mu = 0.000$ のときの結果について考察する．図 4 より，Massive Leave Ratio が増加するに従い，どの方式もダウンロード完了率が低下していることがわかる．特に，提案方式 (proposal) では，ダウンロード完了率が急速に低下している．これは，ウィンドウにより符号化ブロックに含まれるピースが制限されていること，配信開始時には符号化ブロックの重複確率が増大すること (3.4 章参照) から，一斉離脱が起こるまでに，十分な符号化ブロックがネットワーク全体に拡散されないためであると考えられる．これに対し，拡張方式 (proposal2) を用いると，Massive Leave Ratio が 0.4 よりも小さい範囲では，ほとんど全てのピアがダウンロードを完了できている．また，同範囲では，LT 符号方式も，ダウンロード完了率を高く維持できている．一方で，Massive Leave Ratio が 0.6 を超えると，rarest-first 方式が最も高いダウンロード完了率を示している．これは，他方式と比較して多くのピアが，一斉離脱が発生する前にコンテンツを揃えたためであると考えられる．このことは，図 5 に示す，平均ダウンロード完了時間からも示唆される．図 5 より，Massive Leave Ratio に関わらず，Rarest-First 方式が最も平均ダウンロード完了時間を小さくできていることがわかる．一方で，proposal, LT 符号方式は，比較的ダウンロード完了時間が大きい．これは，符号化によるオーバーヘッドの増加と，符号化ブロックの重複による帯域の浪費が原因であると考えられる．これに対し，proposal2 は，ネットワーク参加直後は Rarest-First 方式と同様に直接ブロックを交換することで，符号化ブロックの重複を回避している．これにより，符号化によるオーバーヘッドによるダウンロード完了時間の増加は認められるものの，Rarest-First 方式と比較し，その増分を小さく抑えることができている．

図 6, 7 より，離脱率 $\mu = 0.001$ の場合でも，全体の傾向には大きな変化が見られない．ただし，LT 符号方式では，離脱率 $\mu = 0.000$ のときと比較し，ダウンロード完了率の低下率が大きい．これは，LT 符号はダウンロード完了時間が比較的大きいことから，ピアの離脱までにダウンロードが完了しないピアが存在するためであると考えられる．一方，proposal2 では，LT 符号化方式と比較してダウンロード完了時間を小さくできているため，ダウンロード完了率を高く維持することができている．

以上の結果より，Rarest-First 方式はダウンロード完了時間，LT 符号方式は離脱耐性に優れていることがわかる．これらの方式と比較し，拡張方式は離脱率 μ，及び Massive Leave Ratio が変化した場合であっても，LT 符号方式と同等以上の離脱耐性を獲得しつつ，Rarest-First 方式に近い配信能力を備えることができている．この結果から，拡張方式は通常の離脱，一斉離脱の双方について，離脱耐性を高めることができたといえる．

図 8, 9 に，Massive Leave Ratio がそれぞれ 0.0, 0.4 の場合の，w_1 の選択確率 $P(w_1)$ による拡張方式のダウンロード完了率の変化を示す．ただし，離脱率は 0.001 である．ここで，シミュレーション条件より，コンテンツは 64 個のピースから構成されているので，$k_2 = 64$ のときは，コンテンツ全体を対象にして拡張方式を適用した場合に相当する．図 8 より，一斉離脱が発生しない場合，ウィンドウサイズの設定，及び w_1 の選択確率 $P(w_1)$ は，離脱耐性に影響を与えないことがわかる．一方，40%のピアが一斉に離脱する場合 (図 9)，ウィンドウサイズの設定に関わらず，$P(w_1)$ が大きくなるにつれ，ダウンロード完了率が低下する傾向にある．また，ウィンドウサイズ k_1 が小さいほどダウンロード完了率の低下率が大きく，k_2 の値による影響が大きくなることがわかる．以上のことから，多数のピアが一斉に離脱する環境においては，k_1 の値を大きくし，多数のピースを符号化した方が，安定して離脱耐性を高めることができると考えられる．なお，ダウンロード完了時間にはほとんど差が見られなかったため，割愛する．

5. おわりに

本稿では，離脱耐性の高い P2P 型配信システムの実現を目指し，冗長符号化の一種である Rateless Coding を用いたデータ拡散手法を提案した．特に，符号化効率と完全なピースの入手確率のバランスを考慮し，ピースの希少性に基づき選択された複数のピースに対し，希少なデータの入手確率が向上するように符号化を行った．本提案手法を計算機シミュレーションで評価した結果，従来の Rarest-First 方式と比較し，より多様なピアの離脱に対して，離脱耐性を向上できることを確認した．

今後の課題としては，Tit-For-Tat 戦略等も含めた，Bit-

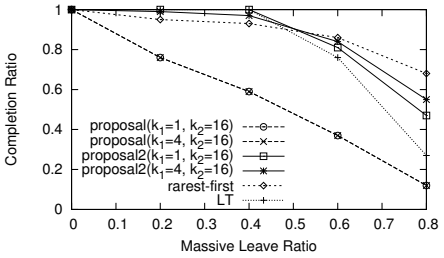

図 4 ダウンロード完了率
(離脱率 μ: 0.000)

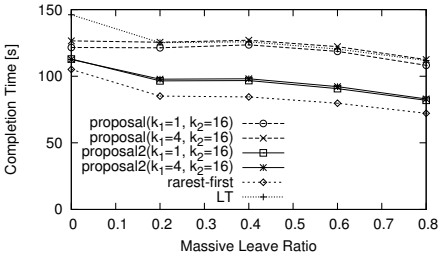

図 5 平均ダウンロード完了時間
(離脱率 μ: 0.000)

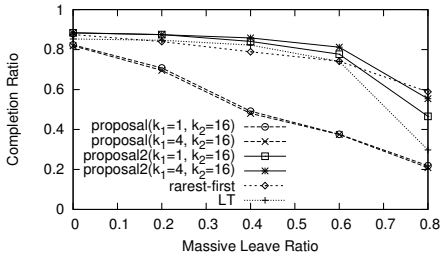

図 6 ダウンロード完了率
(離脱率 μ: 0.001)

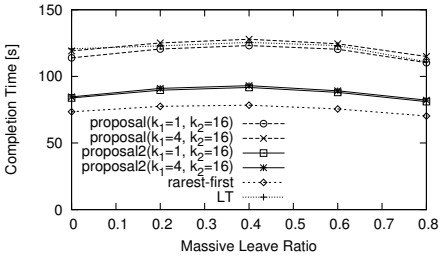

図 7 平均ダウンロード完了時間
(離脱率 μ: 0.001)

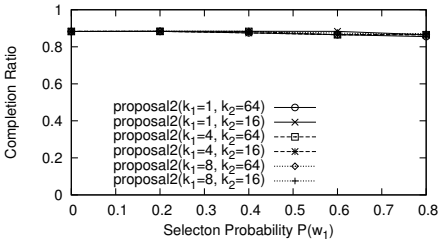

図 8 w_1 の選択確率 $P(w_1)$ による
ダウンロード完了率の変化
(Massive Leave Ratio: 0.0)

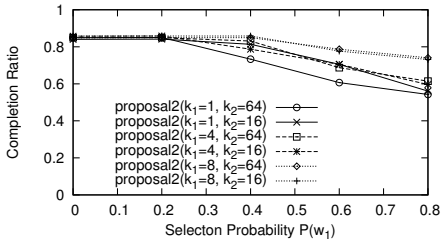

図 9 w_1 の選択確率 $P(w_1)$ による
ダウンロード完了率の変化
(Massive Leave Ratio: 0.4)

Torrent を完全に模した環境における性能評価が挙げられる．また，各ピアが，周囲の環境に応じて，自律分散的にパラメータをチューニングする手法の開発も課題である．

参考文献

[1] BitTorrent, `http://www.bittorrent.com`
[2] The BitTorrent protocol specification, `http://www.bittorrent.org/beps/bep_0003.html`.
[3] Spoto, S., Gaeta, R., Grangetto, M. and Sereno, M.: BitTorrent and fountain codes: friends or foes?, in *Proc. IEEE IPDPSW 2010*, 8 pages (2010).
[4] MacKay, D.J.C.: Fountain codes, *IEE Proceedings-Communications*, Vol. 152, No. 6, pp. 1062–1068 (2005).
[5] sandvine: Global internet phenomena report: 1H 2014, `https://www.sandvine.com/downloads/general/global-internet-phenomena/2014/1h-2014-global-internet-phenomena-report.pdf`.
[6] Gkantsidis, C. and Rodriguez, P.: Network coding for large scale content distribution, in *Proc. IEEE INFOCOM 2005*, Vol. 4, pp. 2235–2245 (2005).
[7] Wu, C. and Li, B.: rStream: resilient and optimal peer-to-peer streaming with rateless codes, *IEEE Trans. PDS*, Vol. 19, No. 1, pp. 77–92 (2008).
[8] Xu, J., Zhao J., Wang, X., Xue, X.: Swifter: chunked network coding for peer-to-peer content distribution, in *Proc. IEEE ICC 2008*, pp. 5603–5608 (2008).
[9] Chou, P. A., Wu, Y. and Jain, K.: Practical network coding, in *Proc.41st Annu. Allerton Conf. Communication Control and Computing*, Vol. 41, No. 1, pp. 40–49 (2003).
[10] Luby, M.: LT codes, in *Proc. IEEE FOCS 2002*, pp. 271–280 (2002).
[11] Sejdinovic, D., Vukobratovic, D., Doufexi, A., Senk, V. and Piechocki, R. J.: Expanding Window Fountain Codes for Unequal Error Protection, *IEEE Trans. Comm.*, Vol. 57, No. 9 (2009).
[12] Guo, L., Chen, S., Xiao, Z., Tan, E., Ding, X. and Zhang, X.: A Performance Study of BitTorrent-like Peer-to-Peer Systems, *IEEE J. Sel. Area, Comm.*, Vol. 25, No. 1, pp. 1–15 (2007).