

修士論文

ピアの churn に対する Tit-for-Tat 型 P2P ファイル配信のロバスト性評価

香山 侑槻

2018 年 2 月 1 日

奈良先端科学技術大学院大学
情報科学研究科

本論文は奈良先端科学技術大学院大学情報科学研究科に
修士(工学) 授与の要件として提出した修士論文である。

香山 侑槻

審査委員：

笠原 正治 教授 (主指導教員)

藤川 和利 教授 (副指導教員)

笹部 昌弘 准教授 (副指導教員)

ピアの churn に対する Tit-for-Tat 型 P2P ファイル配信のロバスト性評価*

香山 侑槻

内容梗概

P2P ファイル配信におけるフリーライダー問題の抑制には、ピア間のピースの交換を促す Tit-for-Tat (TFT) 戦略が有効であることが知られているが、TFT 型 P2P ファイル配信の最適性については十分に明らかにされていない。そこで本論文では、ピアの途中離脱と途中参加で構成される churn に対する TFT 型 P2P ファイル配信のロバスト性を分析する。具体的には、ピアの churn の前後の状況に着目し、それぞれにおけるファイル配信を最適なピースフローの決定問題として定式化する。この問題は整数線形計画問題として定式化できるため、既存の線形ソルバにより最適解を導出できる。数値実験により、まず、1 台のピアの途中離脱のみが発生する場合に着目した場合、途中離脱するピアの保持するピースの希少性の最大値が高いほど、残されたピアの平均ファイル取得時間が劣化することを示す。また、機器の故障やサービス品質へのユーザの不満など離脱の要因によるシステム性能への影響の違いを示す。最後に、ピアの途中離脱と途中参加で構成される churn に対しては、churn の規模、発生時点によらず、churn が生じない場合と比較してシステム全体の平均ファイル取得時間があまり変化しない、すなわち churn に対するロバスト性を有することを示す。

キーワード

Tit-for-Tat 型 P2P ファイル配信, ロバスト性, churn, 平均ファイル取得時間

*奈良先端科学技術大学院大学 情報科学研究科 修士論文, NAIST-IS-MT1651034, 2018 年 2 月 1 日.

Robustness of Tit-for-Tat-Based P2P File Distribution Against Peer Churn^{*}

Yutsuki Kayama

Abstract

It has been pointed out that the free rider problem in P2P file distribution can be suppressed by Tit-for-Tat (TFT) strategy, which encourages peers in exchanging pieces each other. However, the optimality of such TFT-based P2P file distribution has not been fully revealed. In this thesis, we analyze the robustness of TFT-based P2P file distribution against peer churn, which consists of sudden peer departure and arrival. Specifically, we focus on situations before and after peer churn and formulate P2P file distribution in these situations as determination problems of optimal piece flow. Since these problems can be modeled as Integer Linear Programmings (ILPs), optimal solutions can be solved by the existing linear solver. Through numerical experiments, we first show that peer departure with high rarity of pieces degrades the average file retrieval time of remaining peers, when only one peer suddenly departs. We further demonstrate that the system performance changes depending on the factor of peer departure, i.e., system failure or user's dissatisfaction with system performance. Finally, we show that the average file retrieving time of the whole system does not almost change, regardless of the scale and timing of peer churn, which indicates that the TFT-based P2P file distribution is robust against peer churn.

^{*}Master's Thesis, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-MT1651034, February 1, 2018.

Keywords:

Tit-for-Tat-based P2P file distribution, robustness, peer churn, average file retrieving time

目次

1. はじめに	1
2. 関連研究	3
3. ピアの churn を伴う TFT 型 P2P ファイル配信における最適なピースフ ロー決定問題	5
3.1 Churn 発生 の予測困難性を考慮した 2 段階最適化に基づくアプローチ	5
3.2 1 段階目: Churn 発生までの最適化	6
3.3 2 段階目: Churn 発生以降の最適化	9
4. 数値実験	11
4.1 想定環境	11
4.2 ピアの途中離脱による影響	12
4.2.1 1 台のピアが離脱する場合の基本特性	12
4.2.2 障害によりピアが 1 台離脱する場合	16
4.2.3 スループット最小のピアが 1 台離脱する場合	18
4.3 ピアの Churn による影響	19
4.3.1 1 台のピアの churn が発生する場合の基本特性	20
4.3.2 Churn の規模による影響	24
5. まとめと今後の課題	26
謝辞	29
参考文献	30

目 次

1	離脱時点と平均ファイル取得時間の関係（途中離脱ピアのアップロード容量3）	13
3	離脱時点と平均ファイル取得時間の関係（途中離脱ピアのアップロード容量1）	13
2	離脱時点と平均ファイル取得時間の関係（途中離脱ピアのアップロード容量2）	14
4	途中離脱ピアが保持するピースの希少性の最大値と平均ファイル取得時間との関係（途中離脱ピアのアップロード容量3）	15
5	途中離脱ピアが保持するピースの希少性の最大値と平均ファイル取得時間との関係（途中離脱ピアのアップロード容量2）	16
6	途中離脱ピアが保持するピースの希少性の最大値と平均ファイル取得時間との関係（途中離脱ピアのアップロード容量1）	17
7	離脱時点と平均ファイル取得時間の関係（障害による途中離脱）	17
8	離脱時点，アップロード容量と途中離脱ピアとして選ばれる割合との関係	18
9	ピアの不満による途中離脱と障害による途中離脱との比較	19
10	churn 発生と途中離脱のみが生じる場合の比較（アップロード容量3）	20
11	churn 発生と途中離脱のみが生じる場合の比較（アップロード容量2）	21
12	churn 発生と途中離脱のみが生じる場合の比較（アップロード容量1）	21
13	参加時点とその時の参加ピアのファイル取得完了時間の関係（途中参加ピアのアップロード容量3）	22
14	参加時点とその時の参加ピアのファイル取得完了時間の関係（途中参加ピアのアップロード容量2）	23
15	参加時点とその時の参加ピアのファイル取得完了時間の関係（途中参加ピアのアップロード容量1）	23
16	ピア全体の平均ファイル取得時間とファイル配信期間	25
17	ピア群を分類したファイル取得完了時間	27

1. はじめに

Linux や FreeBSD などのフリー OS のディスクイメージ、あるいはソフトウェアのセキュリティ対策用パッチデータについては、大容量のファイルや多数のユーザへの配信を伴うことから、従来のクライアント・サーバ型配信システムではアクセス集中によりファイルサーバの回線容量がボトルネックとなる傾向がある。特に、新しいバージョンの OS やソフトウェアがリリースされた直後は、短期間にユーザからのリクエストがファイルサーバに集中する、フラッシュクラウドと呼ばれる現象が生じることが多い [1, 2]。このような負荷の増大に対して適応力を持つシステムとして Peer-to-Peer (P2P) ファイル配信が挙げられる。P2P ファイル配信システムでは、ファイルはピースと呼ばれる単位に分割され、サービスに参加中のユーザ端末（ピア）が取得したピースを他のピアに提供する。このことにより、参加するピア数に応じた配信能力を有したファイル配信を実現することができる。例えば、Windows 10 では更新ファイルの提供に P2P 配信技術の利用を可能としている [3]

しかしながら、P2P ファイル配信では、個々のピアは所有者が異なるユーザ端末として動作する。ピアにとって、ファイル取得のためのピース受信は必須である一方、他のピアへのピースの送信はそうではない。ピースの送信には自身のアップロード容量を消費することから、ファイル配信以外に利用中の他のネットワークサービスの品質が低下する可能性がある。その結果、ピースを取得するのみで、他のピアにピースを提供しないフリーライダーと呼ばれるピアがシステム内に生じることが知られている [4]。フリーライダーの増加は、システムにおける配信能力の低下につながるため、フリーライダーの発生を抑制する必要がある。特に、サーバが個々のピアを直接制御することは困難であることから、自律分散的な解決策が求められている。この問題に対し、P2P ファイル配信システムの一つである BitTorrent [5] では、ゲーム理論における Tit-for-Tat (TFT) 戦略を導入している。このことにより、各ピアは自身のピース取得のために他のピアへのピース提供が求められるようになる。

TFT 戦略の導入によりピア間のピース交換が促進される一方で、TFT 型 P2P ファイル配信システムの性能については十分に明らかにされていない。近年、こ

のような TFT 型 P2P ファイル配信システムの性能解析に関する研究が進められている [6] . [6] では、いつ、どのノード（サーバもしくはピア）からどのピアへ、どのピースを送信するかを表すピースフローの決定問題を、整数線形計画問題として定式化している。この定式化では、TFT 戦略など種々の制約条件を満たした上で、ピアのファイル取得開始から取得完了までに要する時間（ファイル取得時間）の平均値を最小化することが目的となっている。整数線形計画問題として定式化された問題は既存の線形ソルバ CPLEX [7] によって最適解を導出できる。[6] では、最適解を分析することで最適なピースフローを生み出すためのサーバのピース配信における戦略を明らかにしている。この知見は、最適な配信に向けたサーバ・ピア間の自律分散制御の実現への応用が期待される。また、最適解の導出はシステムの性能上限を与えるため、このようなアプローチは非常に重要である。

ただし、[6] ではすべてのピアはファイル取得完了までシステムに滞在することを仮定していた。実システムでは、機器の故障やファイル配信のサービス品質に不満を感じたユーザによるアプリケーションの終了などにより、ファイル取得の途中に一部のピアがシステムから離脱する可能性がある。また、すべてのピアが同時にファイル取得を開始するとは限らないため、配信途中に新たなピアがシステムに参加する可能性もある。一般に、このようなシステム内でのピアの流動は churn と呼ばれる [8]。ピアの途中離脱に関しては、途中離脱したピアへサーバが提供したアップロード容量や途中離脱したピアの配信能力が損失することで、システムに残存するピアのファイル取得効率が低下する可能性がある。一方、ピアの途中参加が生じると、サーバのアップロード容量の消費につながる一方、元々システムに参加しているピアへはピース配信がすでに行われているため、新たに途中参加したピアはサーバからピースを優先的に取得できるとともに、取得したピースをすぐに他のピアと交換できる可能性が高い。

そこで本論文では、このようなピアの churn に対する TFT 型 P2P ファイル配信のロバスト性を平均ファイル取得時間に与える影響の観点から分析する。本論文では、ピアの churn をピアの途中離脱後に新たなピアの途中参加が生じる状況と定義する。一般にピアの churn の発生は事前に予見することが困難であると考

えられるため、この点を考慮した分析方法が必要となる。この点に関しては、ピアの churn 発生の前後で状況を分けると、[6] に倣ってそれぞれの状況を最適化問題として定式化することができる。いずれも整数線形計画問題として定式化できることから、CPLEX を用いた数値実験により churn による影響を明らかにする。まず、一台のピアが途中離脱する状況を想定し、離脱するピアの特性（アップロード容量）、離脱の発生時点、離脱の原因（機器の故障やユーザのサービス品質への不満度）の観点から基本的な特性を明らかにする。次に、churn の規模と発生時点に対する TFT 型 P2P ファイル配信のロバスト性を評価する。

以降では、2 章で関連研究について述べる。次に、3 章でピアの churn を伴う TFT 型 P2P ファイル配信における最適なピースフロー決定問題について述べる。4 章で数値実験に基づく最適なピースフローの分析結果について述べた後、5 章でまとめを述べる。

2. 関連研究

TFT 型 P2P ファイル配信システムについては、その代表例である BitTorrent を中心に性能評価が行われている [9]。[10] では、BitTorrent における TFT 戦略の有用性とインセンティブの影響を実測に基づき示している。[11] では、フリーライダーを抑制するためのインセンティブメカニズムをゲーム理論に基づき提案し、シミュレーションにより評価している。[12] では、Maze という P2P ネットワークにおいて TFT に基づくインセンティブアルゴリズムの検証を行っている。[13] では、BitTorrent におけるピア間のクラスタリング構造の特性による平均ファイル取得時間への影響を実測に基づき示している。

また、BitTorrent をはじめとする P2P ファイル配信におけるピアの流動性について、様々な解析や評価が行われている。[8] では、Gnutella, Kad, BitTorrent の 3 種類の P2P システムにおけるピアの流動性について数値評価が行われている。[14] では、ピア数、参加頻度、残存時間などの要因が churn に与える影響をシミュレーションにより評価している。また、[15, 16] では、システム内のピア数の変動をそれぞれ偏微分方程式、確率過程に基づきモデル化している。

また, churn がシステム性能に与える影響についても様々な検討が行われている. [17] では, ピアの台数や参加頻度, 残存時間などの要素を考慮した流体モデルを用いたシステムのスケーラビリティと性能の分析が行われている. [18] では, 階層的なクラスタ構造を持つ P2P ネットワークにおいて, churn の規模がトポロジ変化の発生頻度に与える影響を分析している. [19] では, Kademlia ベースの P2P プロトコルを用いて, churn が生じる場合の P2P システムの性能評価をシミュレーションと実証実験により明らかにしている. [20] では, 構造化 P2P ネットワークにおいて, churn 発生率が, 送受信データの損失, 損失に伴う再送信に要する負荷, データ取得に要する時間に与える影響を評価している. [21] では, 非構造化 P2P システムにおいてピアの到着率, 離脱率, 接続ノードの選択といったパラメータに着目して churn のモデル化を行うことにより, ネットワークのサイズ, 入次数出次数, 分離確率, 寿命分布の観点で churn の影響を評価している.

churn に対する対策手法については, ピアが保持しているファイルの複製という観点での検討が行われている. [22] では, ピア間でペアを形成し, ペア内で互いのコンテンツをキャッシュすることで, ピアの離脱に対する耐性を向上させる手法が提案されている. [23] では, 公開・検索・維持のためのプロトコルにより構成されるゾーン複製と呼ばれるアプローチにより churn への耐性を実現している. [24] では, ピアのオンライン時間確率を基にしたピアの保持データに対する可用性を向上するための方式が提案されている.

また, ピアの churn が発生しない静的な TFT 型 P2P ファイル配信において, 平均ファイル取得時間を最小化するような最適なピースフローに関する分析が行われている [6]. [6] では, TFT 型 P2P ファイル配信におけるピースフローの決定問題を, 整数線形計画問題として定式化することで最適なピースフローを導出している. さらに最適解を分析することで, 最適なピースフローの実現に必要なとなる, サーバのピース送信戦略とピアのピース取得戦略について言及している.

本論文では, [6] のアプローチを基に, ある時点で一つのピアが離脱した場合を想定し, ピアの churn が発生する前と発生後とで状況を分けた 2 段階の定式化を行うことで, ピアの churn が発生した場合の, 途中離脱しないピアの平均ファイル取得時間への影響とその原因を明らかにする.

3. ピアの churn を伴う TFT 型 P2P ファイル配信における最適なピースフロー決定問題

3.1 Churn 発生 の 予測困難性を考慮した 2 段階最適化に基づくアプローチ

既存研究 [6] ではピアの churn が生じないような状況について検証が行われていたが、実システムにおいては、端末やネットワーク機器の故障、ユーザのサービス品質に対する不満などの理由により、ファイル配信の途中で一部のピアがサービスから離脱する可能性がある。このようなピアの途中離脱が生じると、サーバから途中離脱したピアに提供されたサーバのアップロード回線容量や途中離脱したピアの配信能力が損失するため、残されたピアの平均ファイル取得時間に悪影響を及ぼす可能性がある。一方、ファイル配信中に新たなピアが参加する可能性もある。ピアの途中参加はサーバのアップロード容量の消費につながる一方で、フラッシュクラウドの場合とは異なり、システム内の他のピアがすでに一部のピースを保持しているため、途中参加ピアが迅速にピースを獲得できる可能性がある。本論文では、このようなピアの churn がシステム性能に与える影響を明らかにする。

ここで、実システムではピアの churn の発生をサーバが事前に予想することは困難である。そこで、ピアの churn の発生以前と以降に状況を分けて検討する。まず、サーバがピアの churn を事前に予想できない場合、churn 発生以前の最適なピースフローは既存の定式化 [6] により導出できる。一方、churn 発生以降の最適なピースフローについては、churn 発生以前におけるピースフローがこの最適なピースフローに等しいという制約の下、churn 発生以降の最適なピースフローを新たに導出すればよい。以降では、このような 2 段階の最適化における定式化と分析の流れについて述べる。

3.2 1段階目：Churn発生までの最適化

既存研究 [6] に倣い、TFT 型 P2P ファイル配信システムを離散時間システムとしてモデル化する。システムには 1 台のサーバ 0 とファイルを取得する N_P 台のピア $1, 2, \dots, N_P$ が存在するものとする。なお、サーバが複数存在する場合にはサーバ 0 を複数のサーバからなる仮想サーバとみなすことで以降の定式化は利用可能である。また、サーバとピアを総称してノードと呼び、サーバ、ピア、ノードの集合をそれぞれ $\mathcal{N}_D = \{0\}$, $\mathcal{N}_P = \{1, 2, \dots, N_P\}$, $\mathcal{N} = \mathcal{N}_D \cup \mathcal{N}_P$ とする。各ピアは M 個のピース ($\mathcal{M} = \{1, 2, \dots, M\}$) で構成されるファイルを取得するものとする。ファイル取得中のピアをリーチャ、取得完了したピアをシーダと呼ぶ。ピース送信時のボトルネックはノードのアップロード容量となることを想定する。また簡単化のため、ノード i ($i \in \mathcal{N}$) のアップロード容量 C_i はノード i が 1 時点あたりに送信可能な最大ピース数とする。全てのピースを取得完了したピアはシーダとしてサーバと同様、他のピアに献身的にピースを提供することもできるが、本論文ではシステムがより過酷な環境下に置かれた場合の性能に興味があるため、シーダとなったピアは即座にシステムから離脱するものとする。

次に、TFT 型 P2P ファイル配信システムにおける最適なピースフローを求める問題を最適化問題として定式化するための変数の定義を行う。まず、ピースフローを決定変数 $x_{t,k,i,j}$ ($i, j \in \mathcal{N}, k \in \mathcal{M}, t = 1, 2, \dots, T$) として次式で定義する。

$$x_{t,k,i,j} = \begin{cases} 1, & \text{時点 } t \text{ でピース } k \text{ をノード } i \text{ が } j \text{ に送信,} \\ 0, & \text{それ以外} \end{cases}$$

ここで T はすべてのピアのファイル取得完了を保証する時間を表す。 T の設定方法としては、サーバのみがピース送信を行うと仮定した場合に全てのピアにピースを送信するのに要する時間を用いることが考えられる [6]。また表記の簡単化のため次式を定義する。

$$\mathcal{T} = \{0, 1, \dots, T\}, \quad \mathcal{T}^+ = \{1, 2, \dots, T\}$$

$x_{t,k,i,j}$ を用いることで、時点 t 終了時にピース k をノード i が保持しているなら 1、そうではない場合は 0 となるような 0-1 変数 $z_{t,k,i}$ ($i \in \mathcal{N}, k \in \mathcal{M}, t \in \mathcal{T}$) と、時

点 t 終了時にノード i がリーチャであるなら 1, シーダ（もしくはサーバ）であるなら 0 となる 0-1 変数 $y_{t,i}$ ($i \in \mathcal{N}, t \in \mathcal{T}$) をそれぞれ以下のように表すことができる.

$$z_{t,k,i} = \begin{cases} z_{0,k,i} + \sum_{s=1}^t \sum_{j \in \mathcal{N}} x_{s,k,j,i}, & \text{if } i \in \mathcal{N}_P, \\ 1, & \text{それ以外} \end{cases}$$

$$y_{t,i} = \begin{cases} 1 - \prod_{k \in \mathcal{M}} z_{t,k,i} & \text{if } i \in \mathcal{N}_P, \\ 0, & \text{それ以外} \end{cases}$$

以上の変数を用いることによって, ピアの churn が発生しない場合における, 平均ファイル取得時間の最小化問題は整数線形計画問題 $P^{(0)}$ として以下のように定

式化できる [6].

$$\min \sum_{i \in \mathcal{N}_P} \tau_i, \quad (1)$$

$$\text{s.t. } z_{0,k,i} = 1, \quad \forall i \in \mathcal{N}_D, \forall k \in \mathcal{M}, \quad (2)$$

$$z_{0,k,i} = 0, \quad \forall i \in \mathcal{N}_P, \forall k \in \mathcal{M}, \quad (3)$$

$$x_{t,k,i,j} \in \{0, 1\}, \quad \forall i, j \in \mathcal{N}, i \neq j, \forall k \in \mathcal{M}, \forall t \in \mathcal{T}^+, \quad (4)$$

$$x_{t,k,i,i} = 0, \quad \forall i \in \mathcal{N}, \forall k \in \mathcal{M}, \forall t \in \mathcal{T}^+, \quad (5)$$

$$x_{t,k,i,j} \leq z_{t-1,k,i}, \quad \forall i, j \in \mathcal{N}, i \neq j, \forall k \in \mathcal{M}, \forall t \in \mathcal{T}^+, \quad (6)$$

$$x_{t,k,i,j} \leq 1 - z_{t-1,k,j}, \quad \forall i, j \in \mathcal{N}, i \neq j, \forall k \in \mathcal{M}, \forall t \in \mathcal{T}^+, \quad (7)$$

$$x_{t,k,i,j} \leq y_{t-1,i}, \quad i \in \mathcal{N}_P, \forall j \in \mathcal{N}, \forall k \in \mathcal{M}, \forall t \in \mathcal{T}^+, \quad (8)$$

$$\sum_{j \in \mathcal{N}_P} \sum_{k \in \mathcal{M}} x_{t,k,i,j} \leq C_i, \quad \forall i \in \mathcal{N}, \forall t \in \mathcal{T}^+, \quad (9)$$

$$\sum_{j \in \mathcal{N}} x_{t,k,j,i} \leq 1, \quad \forall i \in \mathcal{N}_P, \forall k \in \mathcal{M}, \forall t \in \mathcal{T}^+, \quad (10)$$

$$\sum_{k \in \mathcal{M}} (x_{t,k,i,j} - x_{t,k,j,i}) = 0, \quad \forall i, j \in \mathcal{N}_P, \forall t \in \mathcal{T}^+, \quad (11)$$

$$\sum_{i \in \mathcal{N}_P} y_{t,i} = 0, \quad (12)$$

$$\tau_i \leq \tau_j, \quad \forall i, j \in \mathcal{N}_P, C_i \geq C_j \quad (13)$$

(1) は目的関数であり、ピア間での平均ファイル取得時間の最小化を表している、 τ_i はピア i のファイル取得時間を表し、以下のようにリーチャでいる期間として表現できる.

$$\tau_i = \sum_{t=t_i}^T y_{t,i} \quad (14)$$

ここで、 t_i はピア i のシステムへの参加時点を表すものとし、上記の定式化では全てのピアがシステム開始時である $t = 0$ にシステムに参加することとしている ($t_i = 0, \forall i \in \mathcal{N}_P$). (2), (3) は時点 $t = 0$ においてサーバのみが全てのピースを所持していることを表している. (4) は、各ノードでは他のノードへのピース送

信が可能となることを示している。(5)ではピースの送信先が自分自身となることを禁止している。(6)は、自分が所持していないピースは送信できない、(7)は、自分が所持していないピースのみ受信の対象とする制約となる。(8)は、ファイル取得完了したピアがシステムから離脱し、以降はピースを送信しないとする制約である。(9)は、1時点で送信可能なピース数は自分の回線容量以下となる制約である。(10)は、同一のピースは高々1台のノードからしか受信しないことを意味する。(11)は、ピア間で TFT 制約を満たすための制約である。(12)は、少なくとも時点 T までに途中離脱しないピア全てがファイル取得を完了することを表す。(13)は、各ピアのファイル取得完了順序が回線容量の降順となることを表す。なお、バイナリ変数の積に関する制約は線形化できることが知られているため、この問題は整数線形計画問題に帰着される。

2段階最適化の1段階目である、ピアの churn が発生するまでの最適なピースフローは、この問題の解である最適なピースフローの内、churn 発生以前のピースフローに等しくなる。

3.3 2段階目：Churn 発生以降の最適化

2段階最適化の2段階目である、ピアの churn が発生した後の最適なピースフローは、1段階目の最適化問題 $P^{(0)}$ における最適なピースフローの内、churn 発生以前のピースフローを固定した上で、churn 発生以降の最適なピースフローを求める問題として捉えることができる。この問題 $P^{(1)}$ は $P^{(0)}$ を以下のように拡張することで定式化できる。

一般に churn は複数回に渡って発生すると考えられるが、まず、1回目の churn に着目する。実システムでは、各ノードは他のノードの生存をハートビート・メッセージの定期的な受信により判断することが多く、そのため、ノードがピアの途中離脱を検出するには一定の時間を要する。また、ピアが新たに参加した場合には、参加ピアと既存ピアが互いの存在とピースの保持状況を獲得するまでに一定の時間を要する。本論文では、これらの時間が1タイムステップと比較し十分短いものとする。1回目の churn が時点 $t^{(1)}$ 終了時に発生するものとし、その際、ファイル取得途中で離脱するピアの集合を $\mathcal{N}_L^{(1)}$ 、途中から参加するピアの集合を $\mathcal{N}_S^{(1)}$

と定義する．ここで，churn 発生後，開始当初からシステムに残存するピアの集合は $\mathcal{N}_P^{(1)} = \mathcal{N}_P \setminus \mathcal{N}_L^{(1)}$ と表現できる．また，1 回目の churn 発生時点の前後で時点の集合をそれぞれ次式のように定義する．

$$\mathcal{T}_B^{(1)} = \{1, 2, \dots, t^{(1)}\}, \quad \mathcal{T}_A^{(1)} = \{t^{(1)} + 1, t^{(1)} + 2, \dots, T\}$$

$P^{(1)}$ の目的関数は，システムへの参加時点によらず，ファイル取得を完了した全てのピアに対する平均ファイル取得時間の最小化と定義できる．

$$\min \sum_{i \in \mathcal{N}_P^{(1)} \cup \mathcal{N}_S^{(1)}} \tau_i \quad (15)$$

ただし， τ_i は (14) で表されるピア i のファイル取得時間であり，ピアの参加時点 t_i は次式で与えられる．

$$t_i = \begin{cases} 0, & i \in \mathcal{N}_P^{(1)}, \\ t^{(1)}, & i \in \mathcal{N}_S^{(1)} \end{cases}$$

次に，ノードとピア間のピース送信に関する制約 (4) は，churn の発生前後，及びピアの途中離脱，途中参加を考慮して以下のように変更する必要がある．まず，

$$x_{t,k,i,j} = x_{t,k,i,j}^{(0)}, \quad \forall i, j \in \mathcal{N}, \forall k \in \mathcal{M}, \forall t \in \mathcal{T}_B^{(1)},$$

により， $P^{(0)}$ における最適なピースフロー $x_{t,k,i,j}^{(0)}$ ($\forall i, j \in \mathcal{N}, \forall k \in \mathcal{M}, \forall t \in \mathcal{T}_B^{(1)}$) を，ピアの churn が発生するまでの最適なピースフローとする制約条件が必要となる．一方，churn 発生以降については，まず，システムに残存するピアについてはピースの送受信が可能となる制約が必要となる．

$$x_{t,k,i,j} \in \{0, 1\}, \quad \forall i, j \in \mathcal{N}_D \cup \mathcal{N}_P^{(1)} \cup \mathcal{N}_S^{(1)}, i \neq j, \forall k \in \mathcal{M}, \forall t \in \mathcal{T}_A^{(1)}$$

一方，途中離脱したピアについては，churn 発生以降ピースの送受信が行えなくなるという制約が必要となる．

$$\begin{aligned} x_{t,k,i,j} &= 0, \quad \forall i \in \mathcal{N}_L^{(1)}, \forall j \in \mathcal{N}, \forall k \in \mathcal{M}, \forall t \in \mathcal{T}_A^{(1)}, \\ x_{t,k,j,i} &= 0, \quad \forall i \in \mathcal{N}_L^{(1)}, \forall j \in \mathcal{N}, \forall k \in \mathcal{M}, \forall t \in \mathcal{T}_A^{(1)} \end{aligned}$$

最後に、途中参加ピアに対しては、初期のピースの所持状況と churn の発生以前においてピースの送受信に参加できない点に関する以下の制約が必要となる。

$$\begin{aligned} z_{0,k,i} &= 0, \quad \forall i \in \mathcal{N}_S^{(1)}, \forall k \in \mathcal{M}, \\ x_{t,k,i,j} &= 0, \quad \forall i \in \mathcal{N}_S^{(1)}, \forall j \in \mathcal{N}, \forall k \in \mathcal{M}, \forall t \in \mathcal{T}_B^{(1)}, \\ x_{t,k,j,i} &= 0, \quad \forall i \in \mathcal{N}_S^{(1)}, \forall j \in \mathcal{N}, \forall k \in \mathcal{M}, \forall t \in \mathcal{T}_B^{(1)} \end{aligned}$$

このようにして得られた最適化問題 $P^{(1)}$ は、 $P^{(0)}$ からの変更箇所がすべて線形式として表現できるため、 $P^{(0)}$ と同様に整数線形計画問題に帰着される。また、複数の時点に渡って churn が発生する場合にも同様の手順で最適化の段数を増やすことで容易に拡張することができる。

4. 数値実験

本章では、ピアの churn に対する TFT 型 P2P ファイル配信のロバスト性を数値実験を通して明らかにする。まず、一台のピアが途中離脱する状況を想定し、離脱するピアの特性（アップロード容量）、離脱の発生時点、離脱の原因（機器の故障やユーザのサービス品質への不満度）の観点から基本的な特性を明らかにする。次に、連続する途中離脱と途中参加で構成される churn に着目し、churn の規模と発生時点がシステム性能に与える影響を明らかにする。

4.1 想定環境

本論文では、churn に対する各ノードの振る舞いに着目した分析を行うため、比較的小規模なシステムとして、1 台のサーバ 0 が 6 台のピア i ($i = 1, 2, \dots, 6$) に 12 個のピースを配信する状況を想定する。なお、ピースフローを表す決定変数 $x_{t,k,i,j}$ はバイナリ変数であるため、システムサイズ（ノード数、ピース数）の増加に伴い、解空間が指数的に増加する。そのため大規模なシステムを直接分析することは困難であるが、システム全体がピア間の連結成分として複数のサブシステムへと分割可能な場合、サブシステム単位での分析に利用できる。また、シス

テムの規模が小規模であることを考慮し、簡単のためノード間の接続は完全グラフであるものとする。なお、ノード間の接続関係を表す制約条件を追加することで、完全グラフ以外の状況についても分析することは可能である。サーバのアップロード容量がボトルネックとなる状況として、各ノードのアップロード容量 C_i を $C_0 = 2, C_1 = C_2 = 3, C_3 = C_4 = 2, C_5 = C_6 = 1$ とする。評価指標は、平均ファイル取得時間とし、最適化問題 $P^{(0)}, P^{(1)}$ の解は線形ソルバ CPLEX により導出する。それぞれの問題に対する最適なピースフローは複数存在する可能性があるため、以降では 100 回の独立した試行に対する平均と 95%信頼区間を示す。

4.2 ピアの途中離脱による影響

4.2.1 1 台のピアが離脱する場合の基本特性

まず、1 台のピアが途中離脱するとした場合、そのピアのアップロード容量と離脱時点の違いがシステム性能に与える影響を評価する。図 1 から図 3 にアップロード容量が 3, 2, 1 のピアが 1 台離脱する場合の、離脱時点毎の平均ファイル取得時間をそれぞれ示す。なお比較対象として、ピアの途中離脱を予見できる場合と途中離脱ピアがファイル配信に参加しない場合の結果も合わせて示す。前者は、2 段階目の最適化問題 $P^{(1)}$ において 1 段階目の最適化の結果を利用しない場合に相当する。一方、後者は途中離脱ピアを除く 5 台のピアのみのシステムに対する 1 段階目の最適化問題 $P^{(0)}$ に対応する。まず、いずれの図においても、ピアの途中離脱が予見できる場合と途中離脱ピアがファイル配信に参加しない場合の結果が等しくなることがわかる。このことから、ピアの途中離脱を予見できる場合、途中離脱するピアにサーバがピースを送信したとしてもシステム全体の性能を改善できる余地はないことがわかる。

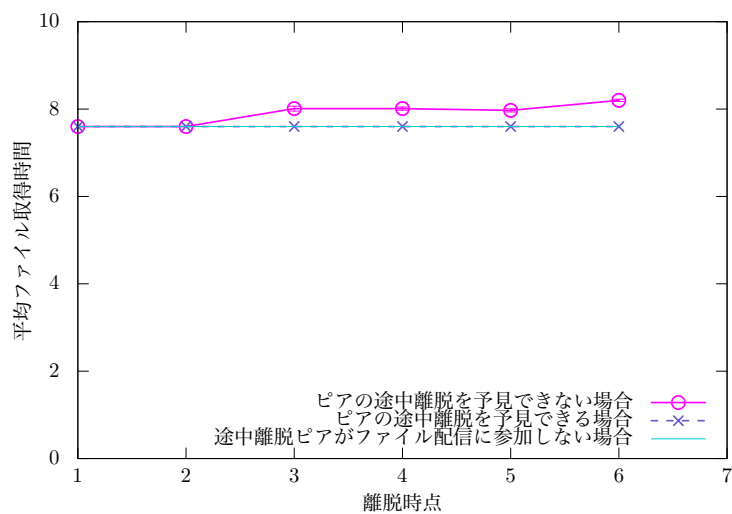


図 1: 離脱時点と平均ファイル取得時間の関係（途中離脱ピアのアップロード容量 3）

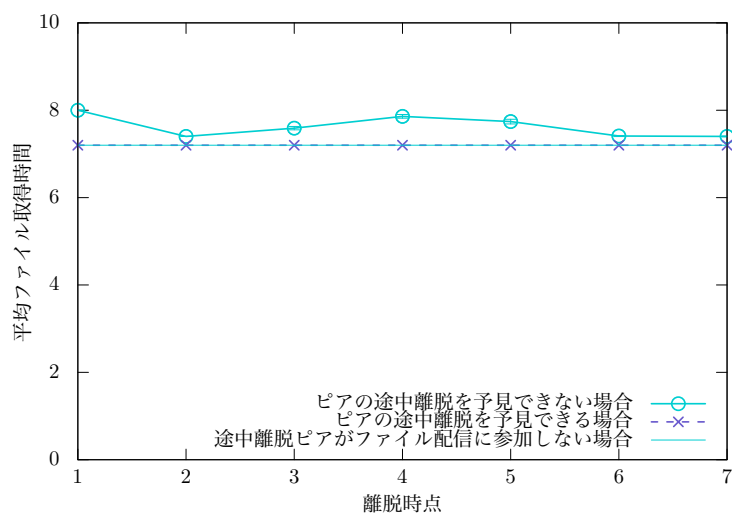


図 3: 離脱時点と平均ファイル取得時間の関係（途中離脱ピアのアップロード容量 1）

次に、ピアの途中離脱が予測できない場合と予測できる場合の結果に着目すると、途中離脱ピアのアップロード容量の違いにより傾向が異なることがわかる。例えば、離脱時点 1 に対しては、途中離脱ピアのアップロード容量が 3, 2 の場

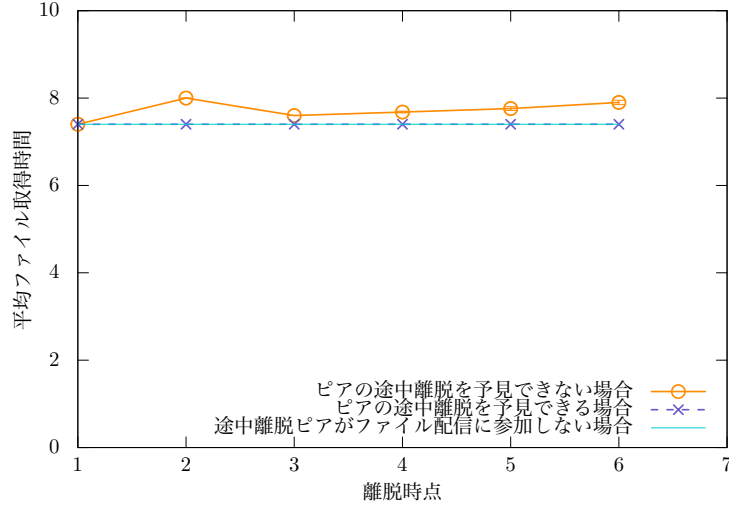


図 2: 離脱時点と平均ファイル取得時間の関係（途中離脱ピアのアップロード容量 2）

合、予見できない場合と予見できる場合とで差がないのに対し、途中離脱ピアのアップロード容量が 1 の場合、予見できない場合の結果は予見できる場合の結果に比べて 11.1%悪化していることがわかる。これは、[6] で示されているピアの途中離脱が生じない場合の最適なピースフローに起因している。前述のように、ピアの途中離脱が予見できない場合、ピアの途中離脱が生じるまでの最適なピースフローはピアの途中離脱が生じない場合のものと等しくなる。この最適なピースフローでは、サーバはピアのアップロード容量の昇順にピースを送信していくことから（以降、アップロード容量昇順送信戦略と呼ぶ）、途中離脱ピアのアップロード容量が 1 の場合、時点 1 でサーバからのピースを受信した直後にピアが途中離脱し、無駄が生じることとなる。

その他の離脱時点に着目すると、途中離脱ピアのアップロード容量によらず、途中離脱を予見できる場合と比べて予見できない場合の結果は 0.00–9.16%悪化していることがわかる。こうした悪化の原因は、システム内のピア間で十分に広まっていないピース、すなわち希少性の高いピースを所持したピアが途中離脱することで、当該のピースに対してサーバが残存するピアに再送する必要が生じる場合があるためと考えられる。なお、ピース $k \in \mathcal{M}$ の希少性は、ピース k を

所持していないピアの割合として定義できる。図4から図6に、途中離脱ピアのアップロード容量が3,2,1の場合に、途中離脱ピアの離脱時点を変化させた場合の、途中離脱ピアの保持しているピースの希少性の最大値と残されたピアの平均ファイル取得時間との関係をそれぞれ示す。いずれの場合にも強い正の相関が確認でき、相関係数は、途中離脱ピアのアップロード容量が3,2,1の場合にそれぞれ0.983,0.907,0.852となった。図中には、線形近似式についても合わせて示しているが、いずれも比較的良好に近似できていることから、途中離脱ピアのアップロード容量とピースの希少性から残存するピアの平均ファイル取得時間を予測できる可能性を示唆している。

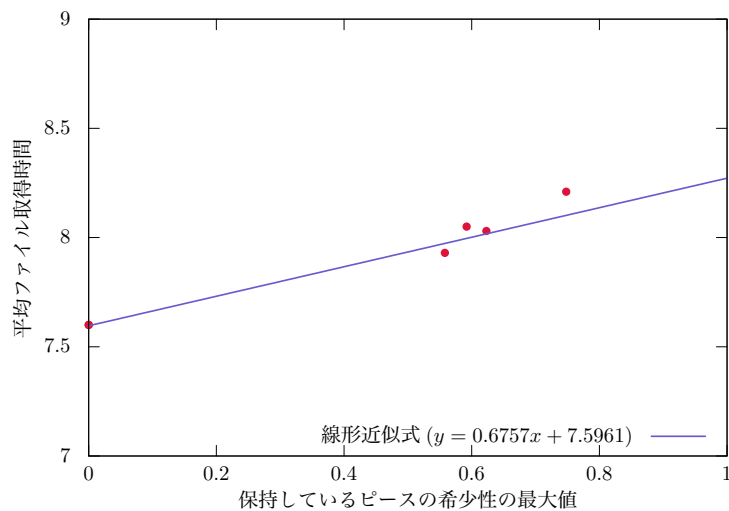


図 4: 途中離脱ピアが保持するピースの希少性の最大値と平均ファイル取得時間との関係 (途中離脱ピアのアップロード容量 3)

実システムでは、どのようなピアがいつ離脱するかを決める要因は複数存在すると考えられる。以降では、端末の故障やシステム障害などの外的要因によってピアが離脱する場合とユーザがファイル取得の進捗に不満を感じることで離脱する場合の2通りについて分析する。

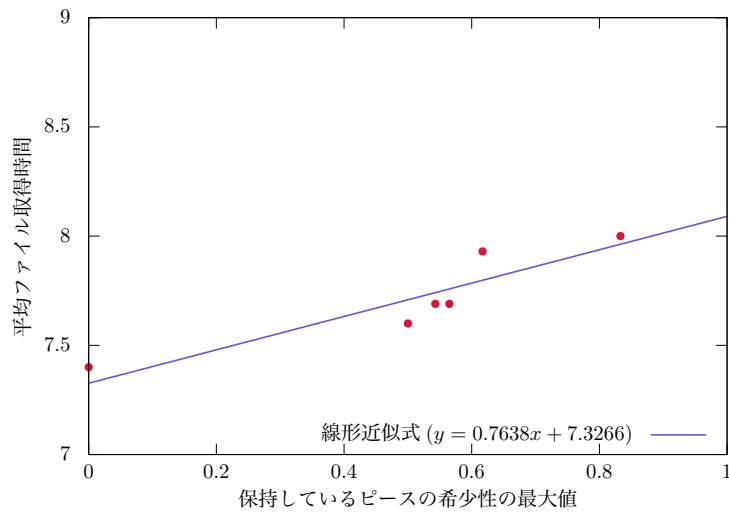


図 5: 途中離脱ピアが保持するピースの希少性の最大値と平均ファイル取得時間との関係 (途中離脱ピアのアップロード容量 2)

4.2.2 障害によりピアが 1 台離脱する場合

まず、端末の故障やシステム障害などの外的要因によってピアが 1 台離脱する場合を想定する。この状況は、1 台のピアがランダムに離脱する状況としてみなすことができるため、図 1 から図 3 における結果に対して、時点毎に単純平均をとることでその挙動を捉えることができる。このようにして得られた、1 台のピアがランダムに途中離脱する場合の離脱時点と平均ファイル取得時間との関係を図 7 に示す。さらに、比較対象として、すべてのノードがピアの途中離脱を予期できるものと仮定した場合の結果を合わせて示す。この結果は、ピアの途中離脱を予期できない場合における性能の上限に対応する。図 7 より、離脱時点が 7 以外の場合には、ランダムな途中離脱の発生により、平均ファイル取得時間が僅かに悪化することがわかる。特に、ファイル配信の中盤から終盤となる時点 4, 5, 6 での途中離脱による影響が 5.68–6.08% の悪化と他の離脱時点での影響よりも大きくなっている。

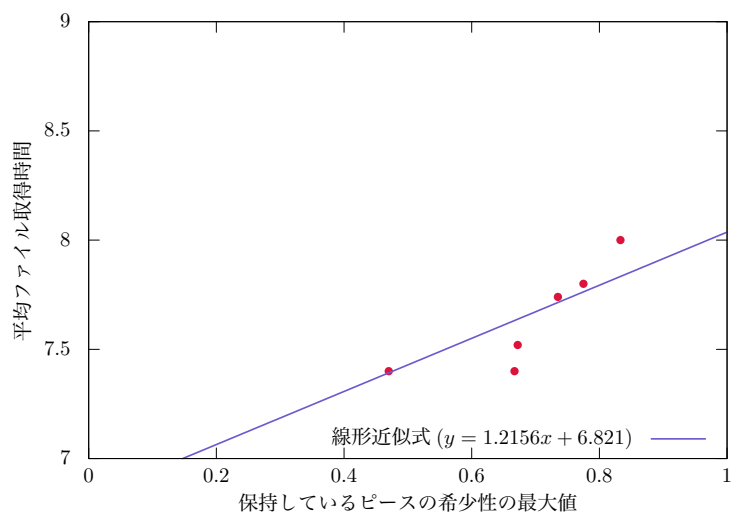


図 6: 途中離脱ピアが保持するピースの希少性の最大値と平均ファイル取得時間との関係 (途中離脱ピアのアップロード容量 1)

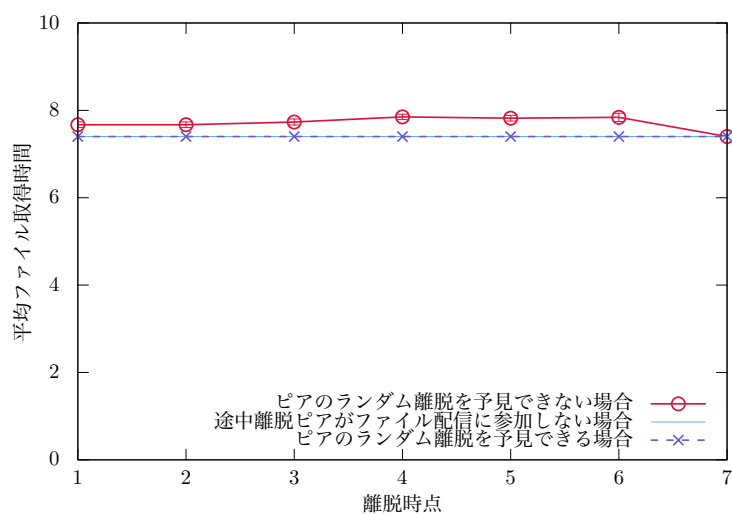


図 7: 離脱時点と平均ファイル取得時間の関係 (障害による途中離脱)

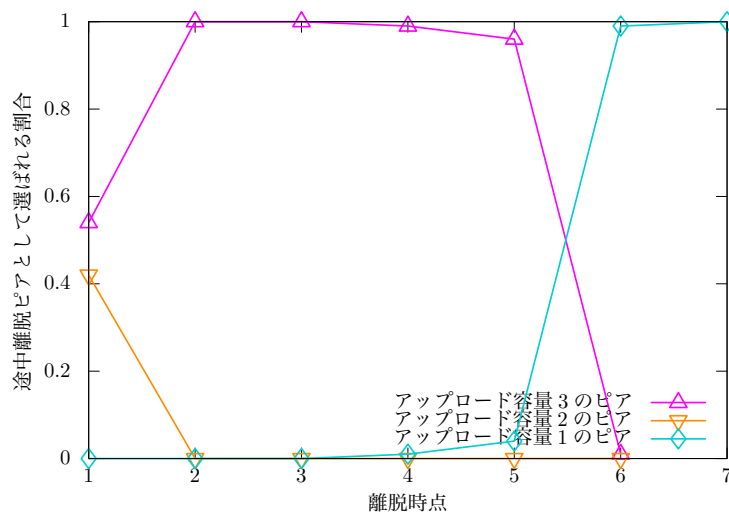


図 8: 離脱時点，アップロード容量と途中離脱ピアとして選ばれる割合との関係

4.2.3 スループット最小のピアが1台離脱する場合

次に，ユーザがファイル配信のサービス品質に不満を感じ，サービスから途中離脱する場合を考える．ファイル配信のサービス品質としては，1時点あたりの平均取得ピース数であるスループットが考えられるが，以降では，ある時点におけるスループットが最小となるピアが1台システムから途中離脱するものとする．その際，スループットが最小となるピアが複数存在する場合はランダムに1台が選択されるものとする．

図8に，離脱時点，アップロード容量と途中離脱ピアとして選ばれる割合との関係を示す．ピアの途中離脱が生じない場合，前述のアップロード用了承順送信戦略に従いサーバがピースを送信することから，序盤ではアップロード容量の高いピアが不満を感じやすい，すなわち途中離脱ピアとして選ばれやすくなっていることがわかる．このようなピアの不満による途中離脱シナリオの性能は，図1から図3における離脱時点毎の平均ファイル取得時間に対して，図8で示される途中離脱ピアとして選ばれる割合により重み付き和をとることで導出できる（図9）．なお図9では，比較対象として前述の障害による途中離脱シナリオの結果を合わせて示している．

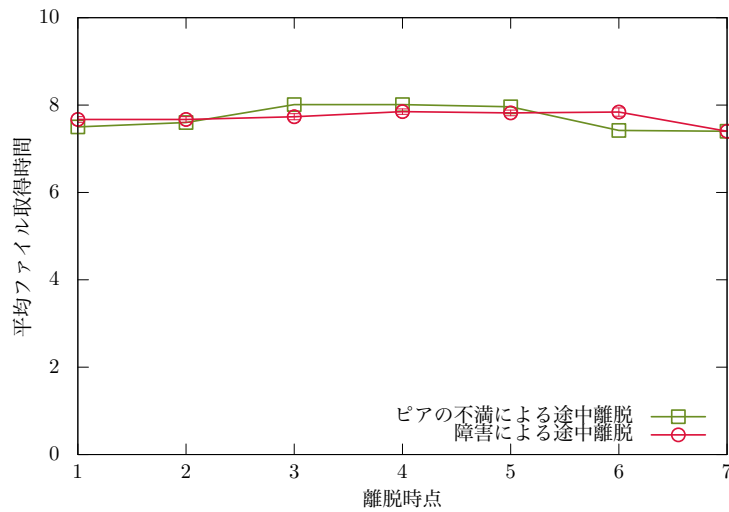


図 9: ピアの不満による途中離脱と障害による途中離脱との比較

図 9 より，離脱時点 1, 2, 6 では障害による途中離脱シナリオの性能が，離脱時点 3, 4, 5 ではピアの不満による途中離脱シナリオの性能がそれぞれ悪くなることがわかる．序盤（離脱時点 1, 2）の挙動は，前述のアップロード容量の昇順にサーバがピースを送信する特性に起因している．中盤（離脱時点 3, 4, 5）では，アップロード容量が大きいピアがサーバからピースを獲得し始めることで保持するピースの希少性が高くなる一方で，低スループットとなる傾向が高いことから，ピアの不満による途中離脱シナリオではそのようなピアが途中離脱ピアとして選ばれやすくなるため，ピアの不満による離脱シナリオの方が障害による途中離脱シナリオより 1.79–3.58%性能が悪化する．終盤（離脱時点 6）では，ピアの不満による途中離脱シナリオでは，ファイル取得に時間がかかるアップロード容量の小さいピアが途中離脱する割合が高くなるため，障害による途中離脱シナリオの方がピアの不満による離脱シナリオより 5.66%性能が悪化する．

4.3 ピアの Churn による影響

4.2 節ではピアの途中離脱に対する TFT 型 P2P ファイル配信のロバスト性を明らかにした．本節では，ピアの途中離脱と途中参加が連続的に発生する churn

による影響を分析する。以降では、まず1台のピアの churn が発生する場合の基本特性を示し、次に、churn の規模に対するシステムのロバスト性について評価する。

4.3.1 1 台のピアの churn が発生する場合の基本特性

ここでは、定常状態ではシステム内のピアの数と構成が変化しないという前提の下、1台のピアがある時点 $t^{(1)}$ で途中離脱した直後に、途中離脱したピアと同じアップロード容量を持つピアがシステムに途中参加するという churn を想定する。

まず、churn による新たなピアの参加が既存の5台ピアの平均ファイル取得時間に与える影響を明らかにする。新たなピアの参加はサーバからの配信先の増加という観点では既存のピアに悪影響を与えるが、途中参加ピアのファイル配信への参加によりピア間のピース交換が活性化される可能性もある。図10, 11, 12

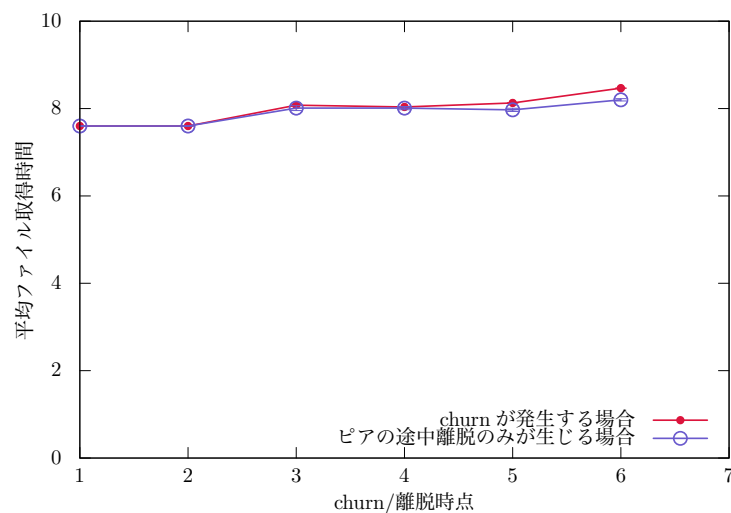


図 10: churn 発生と途中離脱のみが生じる場合の比較 (アップロード容量3)

に、churn 発生シナリオとピアの途中離脱シナリオにおいて、途中離脱（参加）ピアのアップロード容量を3,2,1とした場合の、既存の5台のピアに対する平均ファイル取得時間と離脱 (churn) 時間との関係を示す。図より、churn 発生シナリオと途中離脱シナリオは近い値を示しており、1台の churn の発生は既存ピアの平均ファイル取得時間に与える影響は小さいことがわかる。

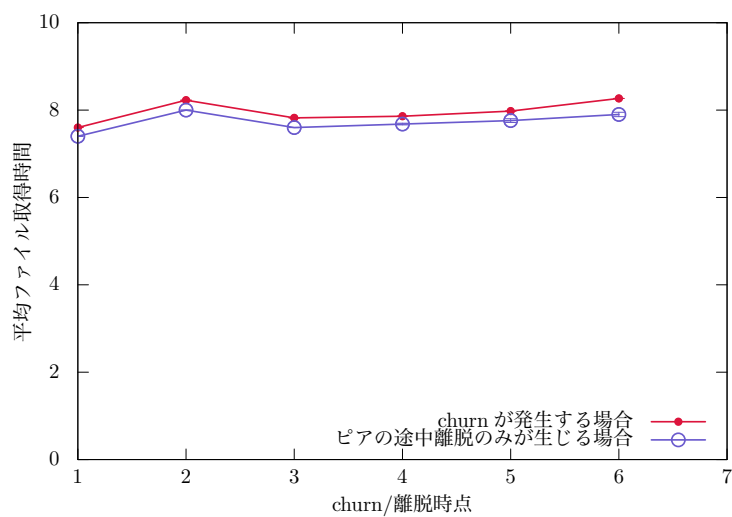


図 11: churn 発生と途中離脱のみが生じる場合の比較 (アップロード容量 2)

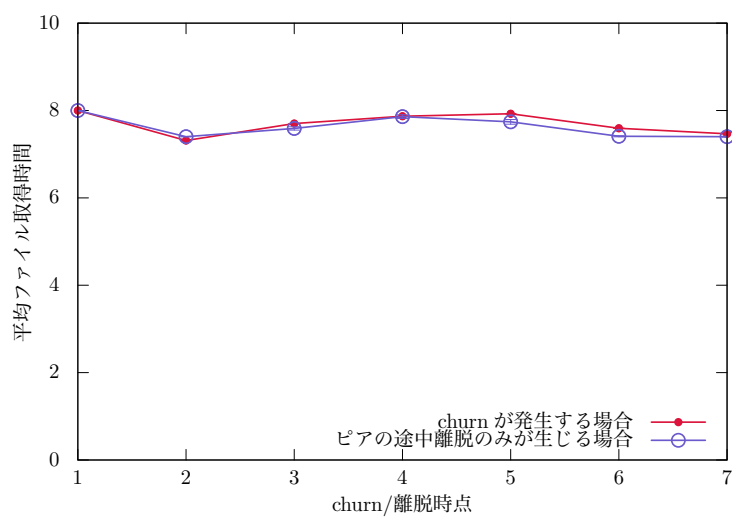


図 12: churn 発生と途中離脱のみが生じる場合の比較 (アップロード容量 1)

次に、churn シナリオにおける途中参加ピアの平均ファイル取得時間に着目する。比較対象として、途中離脱シナリオにおける既存ピアの中で churn シナリオでの途中参加ピアと同じアップロード容量を持つピアの平均ファイル取得時間を用いる。図 13, 14, 15 に、途中参加ピアのアップロード容量を 3, 2, 1 とした場合

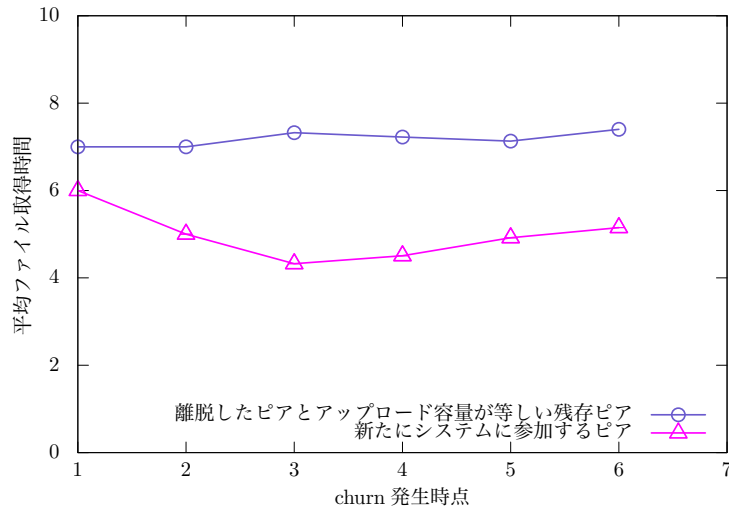


図 13: 参加時点とその時の参加ピアのファイル取得完了時間の関係 (途中参加ピアのアップロード容量 3)

の、途中参加ピアと比較対象のピアに対する平均ファイル取得時間と離脱 (churn) 時点との関係を示す。図より、既存ピアと比較して途中参加ピアの平均ファイル取得時間は 0–41.0% 短い値を示している。これは、途中参加ピアが参加した時点では既存ピア (の一部) がすでにピースを保持していることから、サーバから優先的にピースを取得できるとともに、そのピースをすぐにそれらの既存ピアとのピース交換に利用できるためである。ただし、途中参加ピアのアップロード容量が 1 の場合、サーバのアップロード容量昇順送信戦略では、時点 1 でアップロード容量が 1 のピアにピースが送信された後、当該のピアがシステムから離脱するため、途中参加ピアに対する有利な状況が起こりにくく、結果として離脱 (churn) 時点が 1 の場合での両シナリオ間の性能差はほとんど見られない。

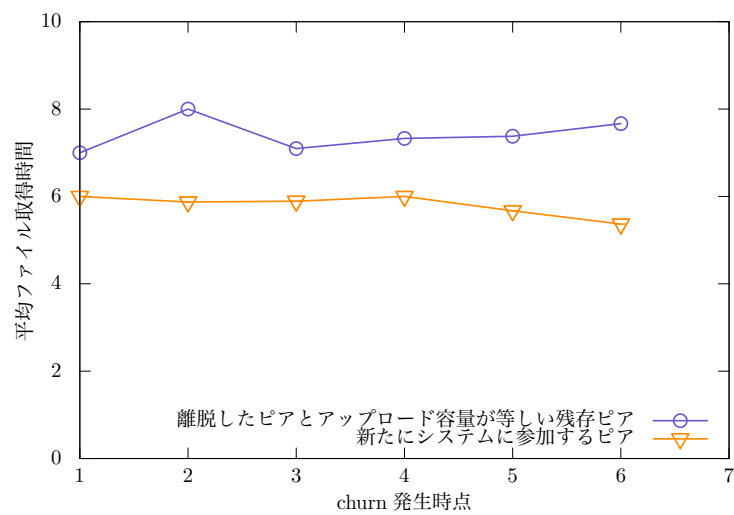


図 14: 参加時点とその時の参加ピアのファイル取得完了時間の関係 (途中参加ピアのアップロード容量 2)

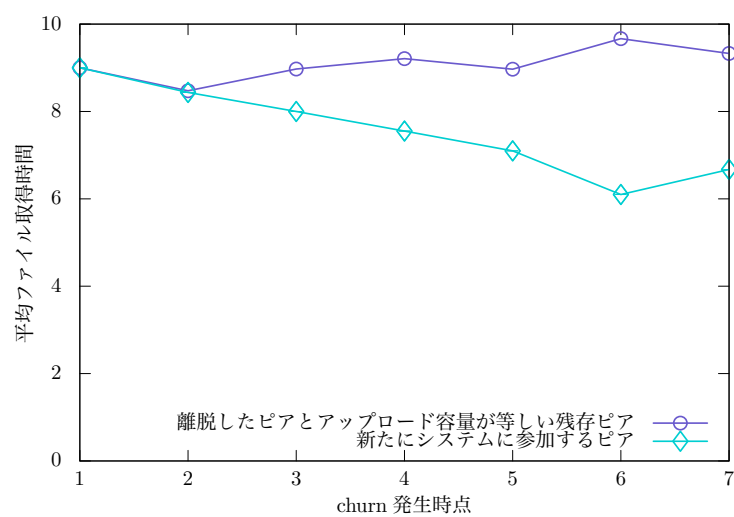
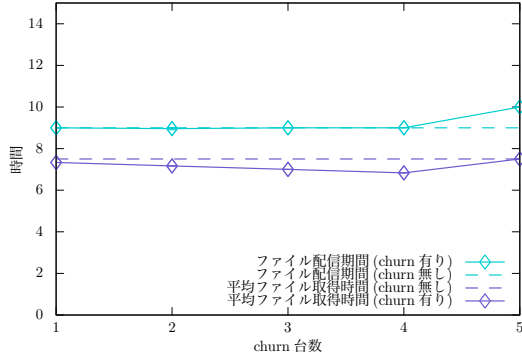


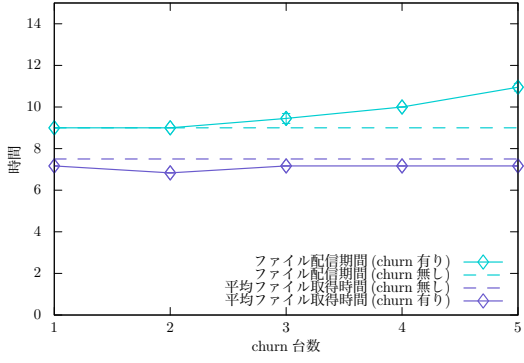
図 15: 参加時点とその時の参加ピアのファイル取得完了時間の関係 (途中参加ピアのアップロード容量 1)

4.3.2 Churn の規模による影響

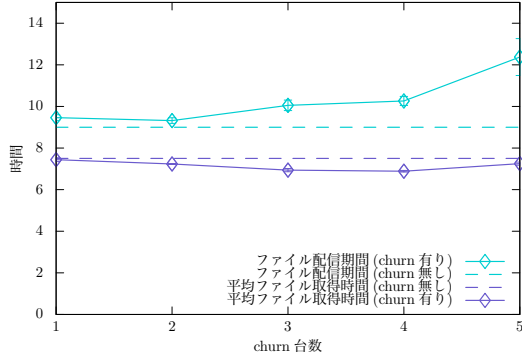
最後に、churn の規模、すなわち途中離脱（参加）するピア数による影響を分析する。以降では、churn の対象となるピアをアップロード容量の降順に選ぶものとする。図 16a–16f に、churn 発生時点毎の、既存ピアの集団と途中参加ピアの集団それぞれに対する平均ファイル取得時間と churn 規模との関係を示す。なお比較のため、churn が生じない場合の結果と、システム開始時点から、既存ピアと途中参加ピアのすべてがファイル取得完了までに要する時間（ファイル配信期間）を合わせて示している。図より、churn の規模、発生時点によらず、多くの場合において churn 有りの場合がなしの場合より平均ファイル取得時間が低くなるものの、両者の差はそれほど大きくないことがわかる。ただし、ファイル配信期間は churn が生じることで、特に churn 発生時点が遅くなるほど増加する点に注意が必要である。



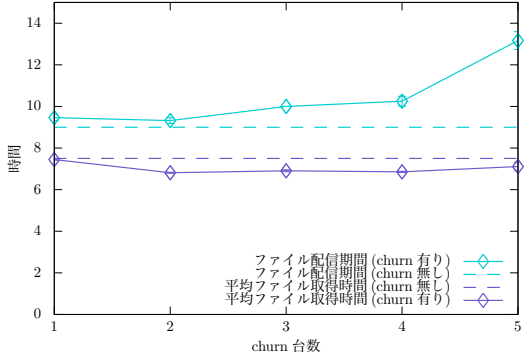
(a) $t^{(1)} = 1$



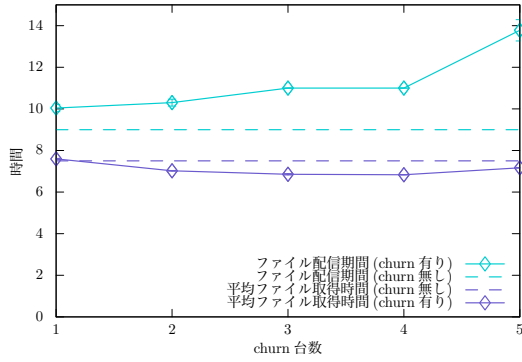
(b) $t^{(1)} = 2$



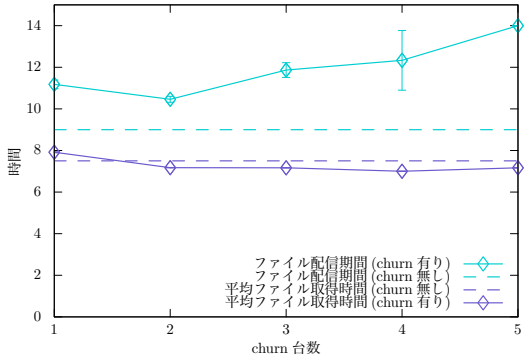
(c) $t^{(1)} = 3$



(d) $t^{(1)} = 4$



(e) $t^{(1)} = 5$



(f) $t^{(1)} = 6$

図 16: ピア全体の平均ファイル取得時間とファイル配信期間

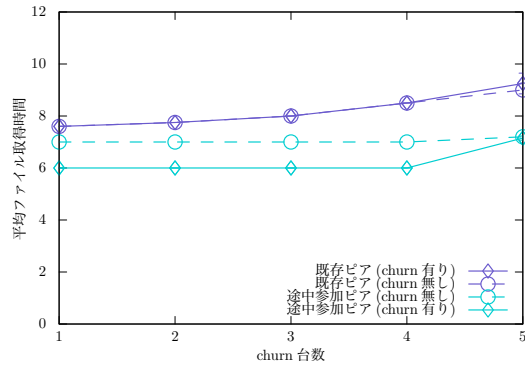
図 17a–17f に、churn 発生時点毎の、既存ピアの集合と途中参加ピアの集合それぞれに対する平均ファイル取得時間と churn 規模との関係を示す。なお比較のため、churn が生じない場合の結果も示しているが、その際、平均ファイル取得時間の計算対象となるピアの集合は churn が生じる場合と同様としている。まず、既存ピアの性能に着目すると、churn が生じない場合と比較して churn が生じる場合は 11.4–22.2% 程度の性能の悪化が見られる。特に、churn の規模が大きくなるほど、また churn の発生時点が遅くなるほど、性能の悪化が大きくなる傾向が確認できる。これは前述のように、churn が生じるとピースを持たない途中参加ピアへのサーバからのピース送信が増えるためである。

一方、途中参加ピアの性能に着目すると、churn の規模が小さいほど性能が良くなる傾向が見られる。これは、途中参加ピアにとっては、同時に参加するピアが少ないほど、サーバから優先的にピースを取得できる機会が多くなるためである。なお、参加時点の観点から見ると、時点 1, 2 といった早期の時点よりもそれ以降の時点の方が性能の改善度合いが大きくなっている。これは、churn の発生時点が遅くなることで、途中参加ピアがシステムに参加した時点で既存ピアがある程度ピースを保持している状況が生まれやすくなるためである。

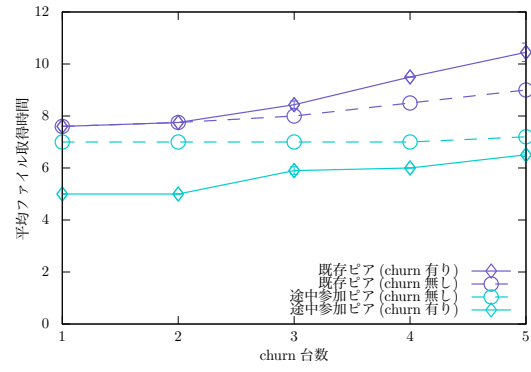
以上のことから、churn が発生しない場合と比較して、churn が発生する場合、churn の規模に対して、既存ピア（途中参加ピア）の平均ファイル取得時間は悪化（改善）する傾向があり、全体として見た場合の性能は僅かに改善することがわかった。このことにより、TFT 型 P2P ファイル配信システムは churn の発生に対して耐性を有していることが示された。

5. まとめと今後の課題

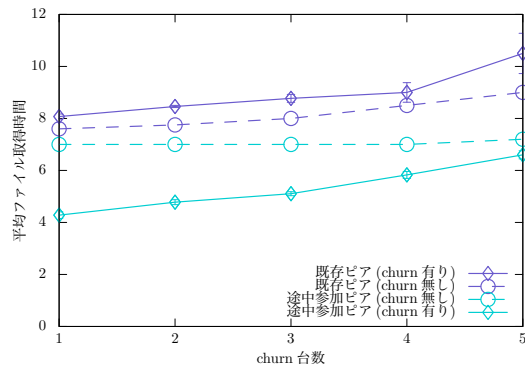
本論文では、ピアの churn に対する TFT 型 P2P ファイル配信のロバスト性を評価した。まず、実システムではサーバがピアの churn が発生する事を予測できない可能性が高いことを考慮し、churn の発生以前と以降に状況を分けた 2 段階最適化のアプローチによりこの問題に取り組んだ。その際、既存研究における、churn の発生しない場合における定式化の知見を用いることで、これらの最適化



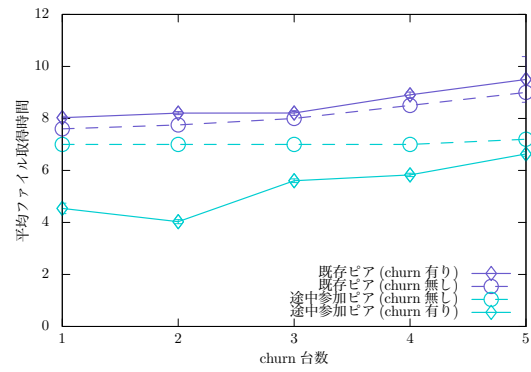
(a) $t^{(1)} = 1$



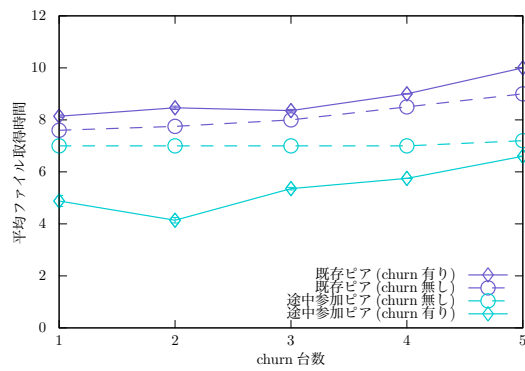
(b) $t^{(1)} = 2$



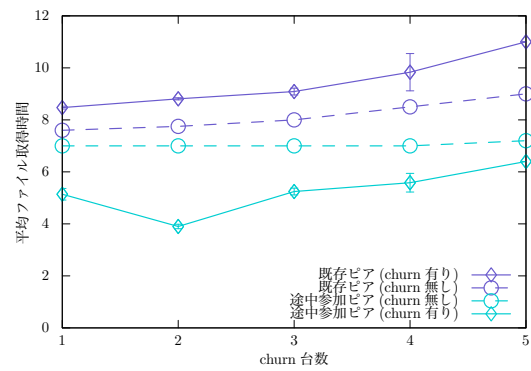
(c) $t^{(1)} = 3$



(d) $t^{(1)} = 4$



(e) $t^{(1)} = 5$



(f) $t^{(1)} = 6$

図 17: ピア群を分類したファイル取得完了時間

問題を定式化できることを示した。

数値実験による評価では、まず、1 台のピアの途中離脱のみが発生する場合に着目し、途中離脱ピアが保持しているピアの希少性の最大値が、残りのピアに対する平均ファイル取得時間との間に強い正の相関を持つことを示した。さらに、システム障害やサービス品質へのユーザの不満などの離脱原因の違いがシステム性能に与える影響についても明らかにした。

最後に、ピアの churn に対する分析の結果、churn が発生しない場合と比較して、churn が発生する場合、churn の規模に対して、既存ピア（途中参加ピア）の平均ファイル取得時間は悪化（改善）する傾向があることがわかった。これは、ピースを持たずにシステムに途中から参加することとなる途中参加ピアに対し、サーバからピースが優先的に送信されることが主な要因となっている。ただし、システム全体として見た場合の性能は、churn が生じる場合の方が生じない場合よりも僅かに改善することがわかった。以上の分析により、TFT 型 P2P ファイル配信システムは churn の発生に対して耐性を有していることが示された。

謝辞

在学中、本研究を続けるにあたりまして、多くの方々にご支援頂きました事をお礼申し上げます。主指導員である、奈良先端科学技術大学院大学 情報科学研究科 笠原正治 教授には、本学における研究の基礎となる知識とご指導と助言を頂きまして心より感謝申し上げます。

副指導員である、奈良先端科学技術大学院大学 情報科学研究科 藤川和利 教授には、学内発表においてご指摘を頂き、研究の助けを頂きました。深く感謝申し上げます。

副指導員である、奈良先端科学技術大学院大学 情報科学研究科 笹部昌弘 准教授には、日頃研究の進捗報告においても様々なご助言ご指摘賜りましたことを深く感謝申し上げます。

奈良先端科学技術大学院大学 情報科学研究科 川原純 助教授には、研究室における様々な技術的学問的な援助を頂きましたことを深くお礼申し上げます。

事務補佐員 橋本洋子 氏、林英子 氏には、本学での様々な事務手続きにおいてご支援頂きまして、また、励ましの言葉も頂きまして誠に感謝しております。

同研究室の学生の皆様には、共同での作業や学業の面に関しても日頃からのご支援頂きました事を感謝致します。

参考文献

- [1] J. Jung, B. Krishnamurthy, and M. Rabinovich, “Flash Crowds and Denial of Service Attacks: Characterization and Implications for CDNs and Web Sites,” Proceedings of the 11th International Conference on World Wide Web, pp.293–304, 2002.
- [2] H. Jamjoom and K.G. Shin, “Persistent Dropping: An Efficient Control of Traffic Aggregates,” Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, pp.287–298, 2003.
- [3] “Optimize Windows 10 update delivery,” <https://docs.microsoft.com/en-us/windows/deployment/update/waas-optimize-windows-10-updates>.
- [4] D. Hughes, G. Coulson, and J. Walkerdine, “Free Riding on Gnutella Revisited: The Bell Tolls?,” IEEE Distributed Systems Online, vol.6, no.6, pp.1–18, 2005.
- [5] “BitTorrent,” <http://www.bittorrent.com/>.
- [6] M. Hasegawa, M. Sasabe, and T. Takine, “Analysis of Optimal Scheduling in Tit-for-Tat-Based P2P File Distribution,” IEICE Transactions on Communications, vol.97, no.12, pp.2650–2657, 2014.
- [7] “IBM ILOG CPLEX Optimizer,” <http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/>.
- [8] D. Stutzbach and R. Rejaie, “Understanding Churn in Peer-to-Peer Networks,” Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement, pp.189–202, 2006.
- [9] R.L. Xia and J.K. Muppala, “A Survey of BitTorrent Performance,” IEEE Communications Surveys Tutorials, vol.12, no.2, pp.140–158, 2010.

- [10] M. Piatek, T. Isdal, T. Anderson, A. Krishnamurthy, and A. Venkataramani, “Do Incentives Build Robustness in BitTorrent?,” Proceedings of 4th USENIX Symposium on Networked Systems Design & Implementation (NSDI 07), 2007.
- [11] S. Jun and M. Ahamad, “Incentives in BitTorrent Induce Free Riding,” Proceedings of the 2005 ACM SIGCOMM Workshop on Economics of Peer-to-peer Systems, pp.116–121, 2005.
- [12] Q. Lian, Y. Peng, M. Yang, Z. Zhang, Y. Dai, and X. Li, “Robust Incentives via Multi-Level Tit-for-Tat,” Concurrency and Computation: Practice and Experience, vol.20, no.2, pp.167–178, 2008.
- [13] A. Legout, N. Liogkas, E. Kohler, and L. Zhang, “Clustering and Sharing Incentives in BitTorrent Systems,” SIGMETRICS Perform. Eval. Rev., vol.35, no.1, pp.301–312, 2007.
- [14] D. Yang, Y.-x. Zhang, H.-k. Zhang, T.-Y. Wu, and H.-C. Chao, “Multi-Factors Oriented Study of P2P Churn,” International Journal of Communication Systems, vol.22, no.9, pp.1089–1103, 2009.
- [15] A. Ferragut and F. Paganini, “Fluid Models of Population and Download Progress in P2P Networks,” IEEE Transactions on Control of Network Systems, vol.3, no.1, pp.34–45, 2016.
- [16] R. Susitaival and S. Aalto, “Modelling the Population Dynamics and the File Availability in a Bittorrent-Like P2P System with Decreasing Peer Arrival Rate,” Lecture Notes in Computer Science, vol.4124, p.34, 2006.
- [17] D. Qiu and R. Srikant, “Modeling and Performance Analysis of BitTorrent-Like Peer-to-Peer Networks,” SIGCOMM Comput. Commun. Rev., vol.34, no.4, pp.367–378, 2004.

- [18] E. Anceaume, R. Ludinard, and B. Sericola, “Analytic Study of the Impact of Churn in Cluster-Based Structured P2P Overlays,” *Proceedings of 2010 IEEE International Conference on Communications*, pp.1–5, 2010.
- [19] Z. Ou, E. Harjula, O. Kassinen, and M. Ylianttila, “Performance Evaluation of a Kademlia-Based Communication-Oriented P2P System Under Churn,” *Computer Networks*, vol.54, no.5, pp.689–705, 2010.
- [20] A. Binzenhöfer and K. Leibnitz, “Estimating Churn in Structured P2P Networks,” *Managing Traffic Performance in Converged Networks*, pp.630–641, 2007.
- [21] Z. Yao, D.B.H. Cline, X. Wang, and D. Loguinov, “Unifying Models of Churn and Resilience for Unstructured P2P Graphs,” *IEEE Transactions on Parallel and Distributed Systems*, vol.25, no.9, pp.2475–2485, 2014.
- [22] X. Meng, X. Chen, and Y. Ding, “Using the Complementary Nature of Node Joining and Leaving to Handle Churn Problem in P2P Networks,” *Computers & Electrical Engineering*, vol.39, no.2, pp.326–337, 2013.
- [23] Z. Trifa and M. Khemakhem, “A Novel Replication Technique to Attenuate Churn Effects,” *Peer-to-Peer Networking and Applications*, vol.9, no.2, pp.344–355, March 2016.
- [24] X. Meng, “A Churn-Aware Durable Data Storage Scheme in Hybrid P2P Networks,” *The Journal of Supercomputing*, vol.74, no.1, pp.183–204, Jan. 2018.