

BitTorrent により配信される音楽や映像ファイルの自動検出

周志強[†] 吉浦紀晃[†]

近年,インターネットの普及に伴って,インターネットのトラフィックが年々増大している.サーバー対クライアントの通信に加えて,クライアント同士が通信する Peer to Peer (P2P)の利用が盛んに利用されるようになってきている.P2P の 1 種である BitTorrent は OS などの大きなファイルのダウンロードに利用される一方,著作権侵害など違法なファイル配信に利用されることがある.本研究では BitTorrent でファイルをダウンロードする時,音楽や映像ファイルを自動で検出し,違法なファイルとしてこのダウンロードを止める仕組みを提案する.

Automatic Detecting Music and Video File Downloaded by BitTorrent

ZHIQIANG ZHOU[†] NORIAKI YOSHIURA[†]

[†] 埼玉大学大学院理工研究科数理電子情報部門情報領域
Department of Information and Computer Science Saitam University

Recently, with internet spreading, network traffic increases year by year. While Server-Client system is used, Peer to Peer (P2P) communication comes to be used actively. BitTorrent which is one kind of P2P is used to download big files such as OS etc. It is also used to download illegal files such as the copyright infringing material etc. This study proposes a mechanism that automatically detects music and video file downloaded by BitTorrent, and stops this download.

1. はじめに

近年のインターネットの普及に伴って,インターネットのトラフィックが年々増大している.全トラフィック中で支配的な割合を占めているのは HTTP ビデオトラフィックと P2P (Peer to Peer) トラフィックである. HTTP トラフィックは, YouTube に代表される動画共有サイトの誕生により急増した. P2P トラフィックの割合は, HTTP トラフィック増加の影響で若干減少しているものの,依然として高いシェアを維持している. 図 1 に ipoque 社[2]が発表したインターネットトラフィックに対する P2P トラフィックの割合を示す[1].

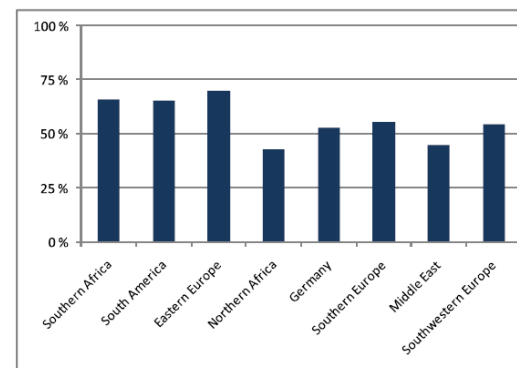


図 1.1 : P2P トラフィックの割合

次に、図2に ipoque 社が発表した P2P トラフィックに対する BitTorrent [3] トラフィックの割合を示す。図を見ると、ファイル共有の手法として普及している BitTorrent プロトコルがトラフィックの大半を占めていることがわかる。

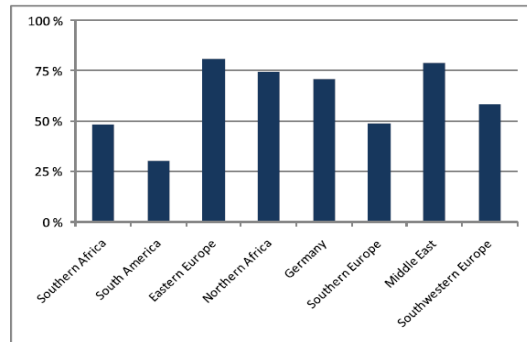


図 1.2:BitTorrent トラフィックの割合

BitTorrent は OS などの大きなファイルのダウンロードに利用されている。しかし、著作権侵害など違法なファイル配信に利用されることがある。世界的に問題となっているインターネット上で行われる違法なアニメと音楽のファイル交換は、BitTorrent 経由だけで 1 週間で 600 万話に達することが明らかになった。これは米国のポップカルチャー情報サイトの ICv2 が、セントラルパーク・メディアのジョン・オダネル社長の発言として伝えている [4]。

上に記述したように、P2P トラフィックに対する BitTorrent トラフィックが帯域の多くを占有している。BitTorrent 経由で違法なアニメと音楽のファイル交換は今後も続く予想される。

現在 P2P ソフトウェアでダウンロードする時、違法なファイルを検出するに関する研究があまりないが、P2P 通信を検知する研究がある。この研究ではネットワーク上に流れる膨大なトラフィックの中から 1 つの通信を構成する複数のパケットの中で、最初のパケットのみを抽出して復号処理する。P2P ソフトの多くは最初の 1 パケット目にユニークな特徴を含んでいる。例えば、Winny なら 1 パケット目の中に含まれる暗号鍵を使って内容を復号すると特定の文字列が現れ、「これは Winny だ」と判断できる。

本論文では、P2P トラフィックの中で最もシェアの高い Bittorrent に着目し、ファイルをダウンロードする時、音楽や映像ファイルを自動で検出し、違法なファイルとしてこのダウンロードを止めることを目的とする。

この目的を達成するため以下 3 つのことを研究目標として定める。

- (1) 早めに音楽や映像ファイルを見つけるアルゴリズムの考案
- (2) 考案した音楽や映像ファイルの発見アルゴリズムの実装
- (3) 実装プログラムのテストとその評価

次章では本論文で扱う P2P と BitTorrent の詳細な説明を行う。第 3 章では早めに音楽や映像ファイルを見つけるアルゴリズムを考案し、第 4 章では開発したプログラムによる実験について述べる。それを受けて第 5 章では実験結果に対する考察を、第 6 章では本論文のまとめと今後の課題について述べる。

2. Peer to Peer(P2P)

2.1 Peer to Peer(P2P) とは

ネットワーク上で対等な関係にある端末間を相互に直接接続し、データを送受信する通信方式であり、また、そのような方式を用いて通信するソフトウェアやシステムの総称である。P2P モデルは、クライアントサーバモデルと異なり自律分散的に動作することで、クライアントサーバ型の通信モデルで問題となるサーバ・ネットワークの負荷集中を避けることができる。

P2P モデルの特徴

P2P は通信形態により性質は異なるが、クライアントサーバモデルと比較すると以下のような特徴がある。

- ピア(直接接続してデータのやり取りを行っているコンピュータ)同士は対等であり、個々のピアが相互に直接接続する。
- ピアの数が増加しても負荷は分散される。
- ピアは低スペックでも動作する。
- ファイル共有を目的とする P2P ソフトウェアの場合、一部のピアがダウンしても別のピアに接続することで動作を続けることができる。
- ピアの管理は困難である。

P2P の通信形態

P2P の通信形態にはピア P2P とハイブリッド P2P、スーパーノード型ハイブリッド P2P がある。

ピア P2P とは、サーバを持たない P2P 方式である (図 2.2 参照)。ピアがインデックス情報 (P2P ネットワークに参加しているピアの情報) を少しずつ分散して保持する。個々のピアがインデックス情報を分散して保持するため、ハイブリッド P2P と比較して検索能力が劣る。代表的なピア P2P のソフトウェアとして、Winny や Gnutella が知られている。

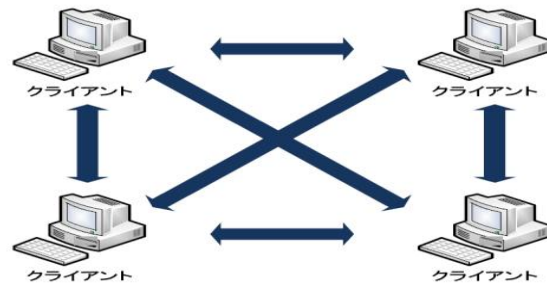


図 2.2 ピア P2P モデルの通信

ハイブリッド P2P とは、データ通信はピア同士で行うが、ピアの検索はサーバに任せる方式である (図 2.3 参照)。BitTorrent、Napster、MSN Messenger はハイブリッド P2P である。

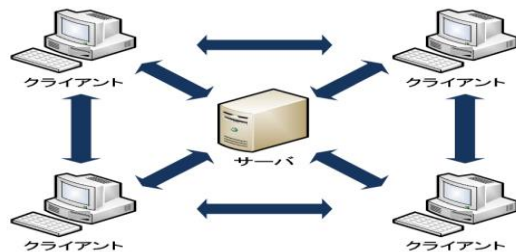


図 2.3 : ハイブリッド P2P モデルの通信

スーパーノード型ハイブリッド P2P とは、処理能力の高いピアをスーパーノードとして選定し、インデックス情報を管理させる方式である (図 2.4 参照)。ピア P2P とハイブリッド P2P 組み合わせた技術である。Skype はスーパーノード型ハイブリッド P2P である。

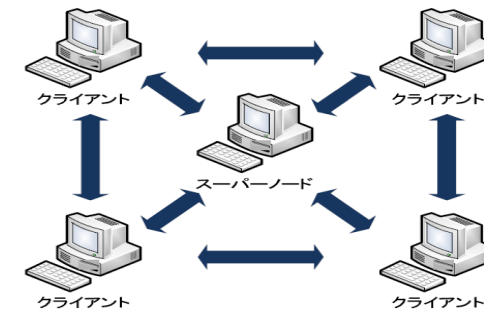


図 2.4 : スーパーノード型ハイブリッド P2P モデルの通信

2.2 BitTorrent とは

BitTorrent とは、Bram Cohen によって開発された、ハイブリッド P2P 方式で動作するファイル転送用プロトコル及びその通信を行うソフトウェアである。ファイル共有のため世界中で利用されており、CentOS や Fedora といった Linux ディストリビューションも BitTorrent で配布されている。

BitTorrent の構成[6]

BitTorrent によるファイル配信の主要構成要素は、トレントファイル、トラッカー、ピアクライアントである。トレントファイルとはダウンロードしようとしているファイルのメタ情報を持つファイルである。トラッカーとは全てのピアとピアのアップロード/ダウンロード、ファイルの取得状況を管理するサーバである。ピアクライアントとは利用者がファイルをダウンロードするためのコンピュータである。

BitTorrent の特徴

- ハイブリッド P2P 方式であり,トラッカーが必要である.
- ファイルを小さいピースに分割してダウンロードやアップロードする.
- Torrent ファイルを入手して,トラッカーからピアの情報を取得し,ピアを選択して,ファイルのダウンロードを行う.
- ダウンロードをする際にはアップロードもしなければならない.

ピアとトラッカーの通信[7][8]

ピアは,HTTP リクエストの GET メソッドを用いて表 2.2 に示す情報をトラッカーに送信する.トラッカーは,応答として表 2.3 に示す情報を送信する.ピアは, interval で指定された秒数が過ぎたら,再びトラッカーにリクエストを送信する.

表 2.2 : ピアがトラッカーに送信する情報

種類	内容
info_hash	共有したいファイルのハッシュ値
peer_id	ピア(BitTorrent クライアント)の ID
port	ピア(BitTorrent クライアント)が使うポート番号
uploaded	アップロードした合計バイト数
downloaded	ダウンロードした合計バイト数
left	ダウンロード残りバイト数

表 2.3 : トラッカーがピアに送信する情報

種類	内容
interval	再びトラッカーにリクエストを送るまでの時間間隔
complete	シーダの数
incomplete	リーチャの数
peers	ファイルを持つピアの一覧

2.3 BitTorrent のファイル転送アルゴリズム[9]

クライアント・サーバモデルでは,ファイルの供給はサーバが行い,クライアントはサーバからファイルをダウンロードする.P2P システムではファイルの供給者と利用者は各 PC に分散され,各 PC はファイルの利用者であると同時にファイルの供給者でもある.

BitTorrent では,「ダウンロードをする際にはアップロードもしなければならない」という規則を導入している.ユーザにアップロードを強制することで,各 BitTorrent クライアントのピアやファイルピース,転送制御のアルゴリズムによって,お互いに効率的にファイルをダウンロードしあうことができる.

2.3.1 ピア選択アルゴリズム

(1)基本的な choke アルゴリズム

BitTorrent では,各ピアはアップロードを一定数のピアに対してアップロードを許可(unchoke)している.デフォルトでは unchoke 状態のピアの数は 4 つである.接続している複数のピアの内, unchoke 状態である 4 つのピアを選択することが重要になる. unchoke 状態であるピアの選択は,基本的には現在のダウンロード速度によって決まる.ダウンロード速度は 20 秒間のダウンロード量から計算し,ピアの選択は 10 秒ごとに行われる.

(2)楽観的な unchoke 戦略

単純に unchoke するピアをダウンロード速度だけで決定すると,もっと良い接続先を見つけないため, unchoke するピアの中で 1 つだけは楽観的な unchoke 戦略をとる.これは,ダウンロード速度を考えずに, 30 秒ごとに自分が知っているピアに順番に unchoke する方法である.

(3)反冷遇主義戦略

ファイルが完全にダウンロードされていない時に,他のピアに貢献しているにもかかわらず,接続している全てのピアから choke 状態にされることがある. 1 分以上 unchoke 状態のピアからピースをまったく取得できなかった場合は,そのピアを choke 状態にして相手のピアに自分は冷遇されていると意志表示する.この場合,楽観的 unchoke 戦略が働いて,代わりの unchoke 状態のピアを見つけ出す.

2.3.2 ピース選択アルゴリズム

ダウンロードするファイルピースの順番は、ダウンロード効率に大きく影響する。例えば、各ピアにいきわたっていないピースを最後にダウンロードしようとする、一斉に他のピアもそのピースをダウンロードしようとして、ファイルの転送効率が落ちる可能性がある。そのため、ピースのダウンロード順序は重要になる。

(1)希少ピース優先戦略

ピースが複数のピアからダウンロードされると、次にどのピースをダウンロードするかが問題になる。BitTorrent では、自分が接続している複数のピアで、各ピアが持っているピースのうちに最も少ないピースをダウンロードしようとする。

(2)ランダム戦略

希少ピース優先戦略が採用されない例外が1つある。それは、BitTorrent クライアントの起動直後である。BitTorrent クライアントの起動直後はアップロードするピースが何もない。そのため、他のピアに一切貢献することができない。他のピアに貢献できるようになるためには、なるべく早く完全なピースを取得して、アップロードできる状態にしなければならない。希少ピース優先戦略を採用すると、希少ピースを持っているピアが少ないため、転送効率が悪くなる。一方、転送効率を優先して、なるべく多くのピアが持っているピースをダウンロードしても、どのピアもそのピースを持っているので、他のピアに貢献することができなくなる。このため、ランダム戦略を採用している。ランダム戦略はダウンロードするピースを完全にランダムに選択する。

3. 音楽や映像ファイル検出方法

BitTorrent は「ダウンロードをする際にはアップロードもしなければならない」という規則を導入している。今のBitTorrent クライアントではダウンロードの効率を考えているため、希少ピース優先戦略、ランダム戦略などピース選択アルゴリズムを使い、ピースを選択する。大抵の場合はダウンロードがピースの順番からではなく、各ピースをランダムな順序でダウンロードする。

本研究ではBitTorrent でファイルをダウンロードする時、音楽や映像ファイルを自動で検出し、違法なファイルとしてこのダウンロードを止める仕組みを提案する。この仕組みではファイルのフレームヘッダからファイルの種類を判定する。ファイルのフレームヘッダは先頭のピースに含まれており、音楽や映像ファイルのフレームヘッダも先頭のピースに含むので、なるべく早めに先頭のピースからダウンロードする必要がある。

ピアクライアントとピアはチョーク状態とアンチョーク状態をもっている。チョーク状態の意味はアップロードを許可されていない状態である。アンチョーク状態の意味はアップロードを許可された状態である。

ピアクライアントとピアの動作

- (1) ピアクライアントはもしピアが持っているピースの中にあるピースを持っていないため、このピアから持っていないピースをダウンロードしたいので、ピアクライアントはこのピアをアンチョークする。もしこのピアもアンチョークしてくれたら、このピアから欲しいのピースのダウンロードができる。逆にこのピアはアンチョークしてくれない、このピアから欲しいのピースのダウンロードができない。
- (2) ピアはピアクライアントが持っているピースの中にあるピースを持っていないため、ピアクライアントをアンチョークする。もしピアクライアントもアンチョークしてくれたら、ピアクライアントからピースのダウンロードができる逆にピアクライアントはアンチョークしてくれない、ピアクライアントからピースのダウンロードができない。
- (3) ピアクライアントは欲しいピースをダウンロードしたい時、まず欲しいピースを持つピアをアンチョークして、もしこのピアはピアクライアントが持っているピースが欲しい、このピアからピアクライアントをアンチョークしなければならない。もしこのピアはピアクライアントが持っているすべてのピースが欲しくない、ピアクライアントは別のピアからいろいろなピースをダウンロードして、欲しいピースを持つピアからアンチョークしてくれるまで、ピースを貯まる。アンチョークしてくれたら、欲しいのピースのダウンロードができる。

3.1 先頭ピースを取るアルゴリズム

- (1) 接続したピアの中から先頭のピースを持つピアを見つけ出す。ピアクライアントはこれらのピアをアンチョークする。
- (2) 先頭のピースを持つピアに先頭のピースのダウンロード要求を出す。
- (3) もし先頭のピースをダウンロードできない場合は、基本的な choke アルゴリズムとランダム戦略のピース選択アルゴリズムを使い、先頭のピース以外のピースを少しダウンロードする。
- (4) また先頭のピースのダウンロード要求を出す。先頭のピースをダウンロードするまで(2)と(3)を繰り返す。

3.2 音楽や映像ファイル検出手法

図 3.3 に示すように、rmb file, wma file, mp4 file, mp3 file 4 種類のファイルの中身を調べ、フレームヘッダは 4 バイト (32 ビット) で出来ている。表 3.2 に示すようにこの 4 種類のファイルのフレームヘッダの特徴をまとめた。

00000000h: 2E 52 4D 46 00	00000000h: 30 26 B2 75 8E
00000010h: 00 07 50 52 4F	00000010h: 9C 04 00 00 00
00000020h: 00 07 EC 7E 00	00000020h: B2 75 8E 66 CF
00000030h: 00 34 F2 D5 00	00000030h: 00 00 00 00 00
00000040h: 00 03 00 09 43	00000040h: 53 00 79 00 6D
rvmb file	wma file
00000000h: 00 00 00 20 66	00000000h: 49 44 33 03 00
00000010h: 69 73 6F 6D 69	00000010h: 00 36 00 00 00
00000020h: 00 03 05 44 6D	00000020h: 77 6E 6C 6F 61
00000030h: 00 00 00 00 7C	00000030h: 3A 2F 2F 77 77
00000040h: 00 04 F1 F3 00	00000040h: 75 73 69 63 2F
mp4 file	mp3 file

図 3.3:音楽や映像ファイルの中身

- (1) ダウンロードする前、音楽や映像ファイルのフレームヘッダを引き出し、分析し、特徴をまとめる。音楽や映像ファイル特徴を保存するため、データベースを作る。
- (2) ファイルをダウンロードする時ダウンロードしようとしているファイルの先頭のピースを読み出す。
- (3) 読み出した先頭のピースの特徴とデータベースにある音楽や映像ファイルのフレームヘッダの特徴を比較する。一致したら、このダウンロードしているファイルは音楽や映像ファイルであると判定する。
- (4) 音楽や映像ファイルであると判定したら、ファイルのダウンロードを止める。

表 3.2:音楽や映像ファイルのフレームヘッダ

ファイルの種類	16 進数	2 進数
mp3 file	49 44 33 03	01001001 01000100 0011001100000011
rmvb file	2E 52 4D 46	00101110 01010010 01001101 01000110
mp4 file	00 00 00 20	00000000 00000000 00000000 00100000
wma file	30 26 B2 75	00110000 00100110 10110010 01110101

4. 実験

この実験では、OS は Linux, CPU は Pentium(R) Dual-Core CPU 2.50GHz, メモリは 1GB の PC を用いた。実験では BitTorrent Client オープンソースソフトウェア [10] 上に提案手法のプログラムの開発を行った。

音楽ファイルのサイズは小さいので、音楽ファイルで先頭のピースを早めに取れるかどうか、確認することが難しい。本論文では映像のファイルを試し、先頭のピースを早めに取れるかどうか実験した。図 4.1 に示すように先頭のピースは 13 番に取れた。先頭のピースを取るまで、時間が 1 分ぐらしかかる。元の BitTorrent Client ソフトウェアを使って、同じ映像のファイルを試し、先頭のピースは 1853 番に取れた。先頭のピースを取るまで、時間が 1 時間以上かかる。2 つの実験を比べて、本論文の提案手法で開発したプログラムは早めに先頭のピースを取れると考えられる。

本論文では、rvmb file, wma file, mp4 file, mp3 file 4 種類のファイルを自動検出の目的で、プログラムを開発した。さらに rvmb 映像ファイルを試して、図 4.1 に示すようにファイルのダウ

ンロードが 0.003%をできた時、ファイルが映像ファイルであることを検出した、ファイルのダウンロードを止めた。

音楽や映像ファイル以外のファイルをダウンロードができるかどうか、確認するため、Fedora ファイルを試して、図 4.2 に示すように、先頭のピースは 20 番にダウンロードできたが、2 時間ぐらいかかって、完全なファイルをダウンロードできた。これでは、本論文の提案手法で開発したプログラムは 4 種類音楽や映像ファイル以外のファイルのダウンロードができると考えられる。

```
[zzq@shulinux bittorrent]$ ./ttorrent Leaf-ToHeart2AD.torrent
metafile size is: 86677
http://nyaatorrents.info:3277/announce
piece length:1048576
get_pieces ok
file_length:201820160
info_hash:f8 7a 02 8f 1b 2b 55 c2 27 18 6f e3 13 fa 98 95 26 6e bd 57
peer_id:-TT1000- 488621136
tracker name:nyaatorrents.info
tracker port:3277
http://tracker.prq.to/announce
tracker name:tracker.prq.to
tracker port:80
request:GET /announce?
info_hash=%f8z%02%8f%1b%2bU%c2%27%18o%e3%13%fa%98%95%26n%bdW&
peer_id=%2dTT1000%2d%20%20%20488621136&port=33550&
uploaded=0&downloaded=0&left=202375168&event=started&
key=48862&compact=1&numwant=200 HTTP/1.0
Host: tracker.prq.to
User-Agent: Bittorrent
Accept: */*
Accept-Encoding: gzip
+++ connecting peer 24.80.172.70 :21722 +++
+++ connecting peer 115.124.237.226 :19894 +++
peer count is :25
+++++ write a slice to btcache index:607 begin:fc000 +++++
##### begin write a piece to harddisk #####
%%% Total piece download:12 %%%
writed piece index:607 total pieces:4289
Complete:0.003% Peers:4 Down:48.00KB/s Up:28.80KB/s
##### end write a piece to harddisk #####
+++++ write a slice to btcache index:1 begin:0 +++++
+++++ write a slice to btcache index:1 begin:4000 +++++
+++++ write a slice to btcache index:1 begin:70000 +++++
+++++ write a slice to btcache index:1 begin:74000 +++++
+++++ write a slice to btcache index:1 begin:fc000 +++++
##### begin write a piece to harddisk #####
%%% Total piece download:13 %%%
writed piece index:1 total pieces:4289
Complete:0.003% Peers:4 Down:60.80KB/s Up:18.29KB/s
##### end write a piece to harddisk #####
This file is a film file.
Connection: closed
```

図 4.1:映像ファイルでの結果

5. 考察

既存の BitTorrent Client では、著作権侵害など違法なファイル配信を利用されている、これは世界中に大きな問題になった。この問題を解消するため、ファイルをダウンロードする時、違法なファイルを検出することは必要である。本論文の提案手法で、音楽や映像ファイルを自動検出することができるが、音楽や映像ファイルは全て違法なファイルではない、また、音楽や映像ファイルの種類が多く存在している。本論文では 4 種類だけ検出できる。これでは不十分と考えられる。今後もっと多くの音楽や映像ファイルの特徴をまとめて、まとめた特徴が作ったデータベースに加入して、開発したプログラムで実験することを考える。

```
[zzq@shulinux bittorrent]$ ./ttorrent Fedora-12-1686-Live-KDE.torrent
metafile size is: 54832
http://torrent.fedoraproject.org:6969/announce
piece length:262144
get_pieces ok
file_length:714081376
info_hash:c4 4b e2 58 58 db ac e0 69 69 65 c6 8b 65 e9 d6 82 8b b7 b4
User-Agent: Bittorrent
Accept: */*
Accept-Encoding: gzip
+++ connecting peer 67.248.68.252 :51376 +++
+++ connecting peer 85.232.158.4 :9889 +++
+++ connecting peer 95.141.193.66 :6881 +++
peer count is :8
+++++ write a slice to btcache index:1 begin:0 +++++
+++++ write a slice to btcache index:2502 begin:18000 +++++
+++++ write a slice to btcache index:1 begin:4000 +++++
+++++ write a slice to btcache index:2502 begin:1c000 +++++
+++++ write a slice to btcache index:1277 begin:8000 +++++
+++++ write a slice to btcache index:1739 begin:30000 +++++
+++++ write a slice to btcache index:1 begin:3c000 +++++
##### begin write a piece to harddisk #####
%%% Total piece download:20 %%%
writed piece index:1 total pieces:2725
Complete:0.007% Peers:6 Down:89.77KB/s Up:0.00B/s
##### end write a piece to harddisk #####
+++++ write a slice to btcache index:1940 begin:0 +++++
+++++ write a slice to btcache index:1086 begin:c000 +++++
+++++ write a slice to btcache index:1739 begin:38000 +++++
##### begin write a piece to harddisk #####
%%% Total piece download:2710 %%%
writed piece index:949 total pieces:2725
Complete:99.45% Peers:5 Down:100.68KB/s Up:0.00B/s
##### end write a piece to harddisk #####
##### begin write a piece to harddisk #####
%%% Total piece download:2725 %%%
writed piece index:1776 total pieces:2725
Complete:100.00% Peers:4 Down:80.25KB/s Up:0.00B/s
##### end write a piece to harddisk #####
```

図 4.2:Fedora での結果

6. おわりに

本論文ではBitTorrentにより配信される音楽や映像ファイルを自動検出するプログラムの開発を行った。先頭ピースを早めにダウンロードできるようにアルゴリズムを考案してプログラムを実装し、テストを行った。その結果、先頭のピースを早めにダウンロードして、音楽や映像ファイルであると判定したら、ダウンロードを止めることができた。

今度、BitTorrent Client オープンソースソフトウェア上にプログラムの開発を行ったが、オープンソースではないBitTorrent Client では、また著作権侵害など違法なファイル配信に利用されることがある。全ての BitTorrent Client に対して、早めに音楽や映像ファイルを自動検出することは今後の課題になる。

参考文献

- [1] YouTube 人気に影響,HTTP トラフィックが P2P を抜く(itmedia)
<http://www.itmedia.co.jp/news/articles/0706/20/news076.html>
- [2] ipoque 社,”Internet Study 2008/2009”
http://www.ipoque.com/resources/internet-studies/internet-study-2008_2009
- [3] BitTorrent 社,”BitTorrent”
<http://www.bittorrent.com/>
- [4] 違法アニメのファイル交換 BitTorrent だけで週 600 万話
<http://animeanime.jp/biz/archives/2007/12/bittorrent600.html>
- [5] ファイル共有ソフトの P2P 通信を 99.78%の精度で検知
日立製作所 システム研究開発所
- [6] BitTorrent プロトコル仕様書
<http://lowlife.jp/yasusii/stories/16.html>
- [7] Cohen,Bram(2003) Incentives Build Robustness in BitTorrent, (2003)
- [8] Rober Layton,Paul Watters:”investigation into extent of infringing content on BitTorrentnet works”Internet Commerce Security Laboratory,pp.8-10 (2010)
- [9] BitTorrent のファイル配信メカニズム
<http://dev.ariel-networks.com/articles/unixmagazine/bittorrent/>
- [10] BitTorrent software development
<http://wenku.baidu.com/view/863a2ad97f1922791688e827.html>