

## P2P ネットワークにおける ストリーミング映像配信システムの実装

村田 修一郎<sup>†1</sup> 藤井 裕史<sup>†1</sup> 坂下 秀<sup>†1</sup>

本稿では、位置情報をもとにした映像を録画し、ライブまたは録画済み映像を検索し P2P で配信を受けるためのシステムの開発・実装について述べる。

システムは P2P フレームワーク PIAX 上に構築された P2P 基盤ソフトウェア musasabi [1] において動作する。位置情報の登録および検索には musasabi の位置情報管理サブシステムを用い、不安定な P2P ネットワーク上で安定した位置情報管理を実現する。

映像情報のストリーミングには、ツリー構造上でのプッシュ方式を採用し、レイテンシの低減と配信の効率化に努めた。また、ストリーミングデータをサブストリームに分割したうえでパリティストリームを加え、配信データの欠損に対応する。

### An Implementation of Media Streaming System on Peer-to-peer network

SHUUCHIROU MURATA,<sup>†1</sup> HIROSHI FUJII<sup>†1</sup>  
and SHIU SAKASHITA<sup>†1</sup>

In this paper, we describe a system, that we have developed, to record video with positional informations and to receive live or recorded media streamings on the peer-to-peer network.

The system is implemented on peer-to-peer-based OS musasabi constructed on the PIAX peer-to-peer framework and uses positional information management subsystem of the musasabi to receive benefit of the fault-tolerance of the musasabi.

We utilize tree-based media streaming network with push method to achieve low latency and high efficiency. Streaming data are divided into substreams and parity stream will be added to these data to avoid decode flaw from lack of the data.

### 1. はじめに

近年、携帯端末の小型化・高性能化、および、携帯電話回線の高速化・低価格化が進んでいる。さらに、GPS (Global Positioning System) やビデオカメラを搭載した携帯端末も普及しつつある。これによって移動端末から位置情報を伴ったストリーミング映像の発信が可能になってきた。

著者らは、多数の人々が移動端末を持って自分が興味ある場所や場面の映像を位置情報付きで自由にストリーミング配信し、またそれを視聴できるシステムの構築を目指している。提案システムでは、位置情報に基づいて映像を検索できる。映像はリアルタイムで閲覧することも、過去の蓄積された映像を閲覧することも可能である。さらに、指定したエリアで将来誰かが映像を撮影したらその映像を視聴するといった視聴予約や、さらに指定したエリアをたまたま通りかかった誰かに何か (視聴者の見たいもの) の撮影を依頼するということも可能である。これらの機能により、利用者は自分の興味ある場所 (被災地や事故現場、行きたい観光地など) の様子を自由に眺めることができる。

著者らの知る限り、既存のメディア (テレビ放送、YouTube のような動画配信サイトなど) でこのような機能を有するものは存在しない。

本システムは多数の参加者 (撮影者および視聴者) を想定している。注目度の高い映像は多数の視聴者が同時に閲覧する可能性があるため、サーバクライアント方式で配信するとサーバが過負荷となる可能性が高い。このため、提案システムは Peer-to-Peer (P2P) 方式で実現する。

この研究は、独立行政法人情報通信研究機構の高度通信・放送研究開発委託研究/新世代ネットワークサービス基盤構築技術に関する研究開発課題エ広域 P2P 型オーバーレイ利用技術の一環としてなされたものである。

### 2. システムの概要

本システムでは、多くの参加者がビデオカメラおよび GPS を備えた携帯端末を持ち歩き、様々な場所で撮影した映像をストリーミング配信する状況を想定している。視聴者はネットワークを介してこれらの映像を受信し視聴する。ストリーミング配信を含め、全ての機能

<sup>†1</sup> アクタスソフトウェア  
Acutus Software

は P2P ネットワーク上で実現する。過去の映像を視聴できるようにするため、撮影した映像は P2P システムを構成する複数のピアに分散して蓄積する。位置情報は携帯端末の GPS から取得しメタデータとして映像に付与する。映像や携帯端末の位置情報の蓄積・検索も P2P システムで行う。

本システムでは、P2P 基盤ソフトウェア musasabi の機能を利用して視聴者が指定した位置と時刻の範囲をもとに映像を検索できる [2]。これは、現在撮影中のライブ映像と蓄積映像が対象である。

### 3. システムの実装

著者らは、移動端末を含む P2P ネットワーク上でのストリーミング映像配信技術の研究開発を行い、以下の技術から構成される配信システムを実装した。

- 能力の低い携帯端末を含む、さまざまな能力を持つノード向けの映像配信を実現するために、複数の解像度での再生が可能な形式に映像を圧縮および変換する技術。元映像を 1/4 解像度の映像、および元の解像度に復元するための差分データに分割する。詳しくは 3.1 節で述べる。
- 変換された映像を蓄積し、P2P ネットワーク上で構成したツリー構造を用いて配信する技術。分割された映像データおよび音声データについてそれぞれツリーを構成し、配信する。詳しくは 3.2 節で述べる。
- 不安定な P2P ネットワーク上で、ノードの離脱等が発生した場合でもストリーミング配信を途切れさせないようにするための技術。分割された映像データおよび音声データに加えて、パリティデータを流すためのツリーを構成する。詳しくは 3.3 節で述べる。

#### 3.1 Motion JPEG を用いた階層化符号化による映像圧縮および解像度変換の実装

映像圧縮の基本的アルゴリズムとしてフレーム内圧縮のみを行う Motion JPEG を採用しつつ、さまざまな能力を持つ視聴ノード向けの映像変換の方式として、階層化符号化により同一映像の複数解像度化機能を持つ映像圧縮方式の開発・実装を行った。

開発は主に Java を用いて行い、実装のベースとして、Java でマルチメディアデータを扱うためのフレームワークのひとつである FMJ [3] を使用した。

##### 3.1.1 映像の複数解像度化機能の実装

映像の複数解像度化機能とは、1 フレーム分の映像データを、それぞれ元の映像データよりも容量の小さい

- 低解像度化した映像データ

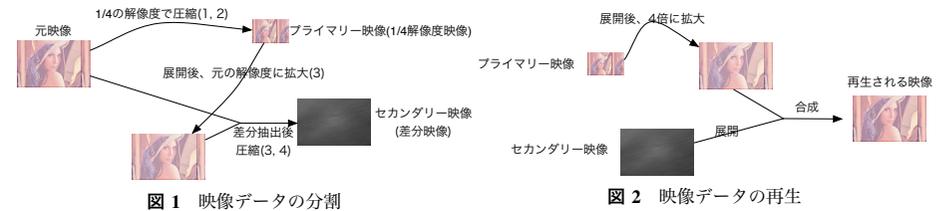


図 1 映像データの分割

図 2 映像データの再生

- もとの解像度に復元するための差分データ

に分割し、映像を表示する側で低解像度データのみを取得して解像度の低い映像を表示することも、両方のデータを取得してもとの解像度で表示することも可能にする機能である。

映像データの分割は以下の手順で行う (図 1) :

- (1) 元の映像データの解像度を縦横それぞれ 1/2 にし、全体として 1/4 化する
- (2) 1/4 化した映像データを JPEG 圧縮する
- (3) 圧縮した映像データを展開し、元の映像データとの差分を取る
- (4) 差分データを JPEG 圧縮する

2 で生成される JPEG 圧縮データを「プライマリー映像データ」、4 で生成される JPEG 圧縮データを「セカンダリー映像データ」と呼ぶ。このふたつのデータを選択的に用いることで低解像度化した映像の表示と元の解像度の映像の再生が可能になる。

低解像度で再生する場合はプライマリー映像データを展開するだけで良く、高解像度で再生する場合には、プライマリー映像データおよびセカンダリー映像データをそれぞれ展開し、合成して表示する。(図 2)

#### 3.2 P2P ネットワーク上でのストリーミング映像配信の実装

P2P ネットワーク上でのストリーミング映像配信システムとして、P2P プラットフォーム musasabi 上に

- 映像を撮影するノード (撮影ノード)
- 映像蓄積システム
- 映像を視聴するノード (視聴ノード)

の三者からなるシステムの開発・実装を行った。

本システムでは、高信頼な映像蓄積システムを実装し、撮影者から送信された映像データを映像蓄積システムに蓄積しつつ、ライブ配信と蓄積したデータの配信を行う。撮影ノードは一般に性能が低くその上り帯域も限られていることから、撮影ノードが直接視聴ノード

にデータを送ることはせず、視聴者への配信は映像蓄積システムが行うことにする。そのため、本システムは全体として以下のふたつのデータ送信をおこなう部分に分解される。

- 撮影ノードから映像蓄積システムへの送信
- 映像蓄積システム (= 配信ルートノード) から視聴ノードへの配信

撮影ノードから映像蓄積システムへの送信は、撮影ノードの負担を軽減するため特別な送信方式を用いず、映像蓄積システムを構成するノードへの直接送信とした。一方、視聴者への配信については、映像蓄積システムが配信ルートとなるツリー型の配信システムを採用し、遅延を短かく保ちつつ全体の転送データ量の低減を計った。

### 3.2.1 ストリーミング映像の蓄積と位置情報システムとの連携

撮影ノードは撮影開始にあたり、まず映像の送信先となる映像蓄積システムを選択する。その際、GPS で取得した位置情報をもとに、PIAX [4] の位置情報オーバーレイ (LL-Net) 機能を用いて近傍の映像蓄積システム探索し、送信先に決定する。

本システムの撮影ノードは QVGA (320 × 240) 15FPS で映像を撮影する。撮影ノードで撮影された映像は Motion JPEG フォーマットで圧縮されたのち、選択された映像蓄積システムに直接送信され、トランスコード (分割および圧縮, 3.1 節参照) ののち保存される。その際、映像蓄積システムは、撮影ノードから送信された映像に対し以下の手順で処理をおこなう。

- (1) 位置情報システム [2] に「ライブ映像」として撮影地の位置情報を登録し、トランスコードを開始する
- (2) トランスコード中は、視聴ノードに対しライブ映像のストリーミングを提供する
- (3) 撮影ノードからの送信およびトランスコードの終了後、位置情報システムに「蓄積映像」として登録しなおす。その際、映像の収録時間も登録する
- (4) トランスコード終了後は、視聴ノードに対し蓄積映像のストリーミングを提供する  
この際、位置情報システムに登録される情報は以下の通りである。

- 映像が撮影された位置
- ライブ映像か蓄積映像か
- 蓄積映像であれば、撮影開始および終了時刻
- 映像を配信可能な映像蓄積システムのネットワーク上の位置

各視聴ノードはこれらの情報をもとに、2 の時点でライブ映像を、また 4 の時点で蓄積映像を視聴することができる。

視聴ノードは以下の手順で、目的の映像をストリーミング可能な映像蓄積システムを発見

し、映像のストリーミングを開始する。

- (1) 視聴ノードは位置情報システムと通信し、目的の映像の撮影位置を含む範囲検索する。位置情報システムは検索結果の映像のリストを返す
- (2) 視聴ノードは映像のリストからひとつを選択し、登録された映像蓄積システムに対して、映像のストリーミングを要求する

### 3.2.2 ツリー型配信による分割された映像データのストリーミング

#### 3.2.2.1 ストリーム分割の利点

映像蓄積システムは同一の映像を視聴する各視聴ノードに対し、ツリー型の配信ネットワークを構成し、映像データのストリーミングをおこなう。<sup>\*1</sup> 本節では、映像蓄積システムを本節での役割に従い「配信ルートノード」と表記する。

3.1 節で述べたように、階層化符号により分割された映像データを選択的に受信することで、各視聴ノードの能力に応じ、異なる解像度の映像を再生することができる。

この機能をツリー構造上での配信で実現するために、ふたつの映像データに音声を加えた計 3 本のデータストリームをそれぞれ配信するツリーを構成する。各ノードは必要なデータを流すツリーのみに加わることで、その能力に応じた映像の再生が可能になる。

ノードごとに受信するデータが異なる最もシンプルなケースとして、配信ルートとなる映像蓄積システム、およびそれぞれ以下のストリームを要求する視聴ノードが、各 1 ノードずつ存在する場合を考える。

- 高解像度の映像および音声を再生する高性能ノード
- 低解像度の映像および音声を再生する低性能ノード
- 音声のみを再生するノード

この場合配信ツリーは図 3 のように構成することができ、ストリームを分割せず個別に最適なデータを配信する場合 (図 4) と比較して配信元ノードの負担を軽減することができる。また、高解像度映像および音声を全てのノードにリレー配信する場合 (図 5) に比べ全体の転送量を減らすことができる。

#### 3.2.2.2 ストリーム分割の実装

本システムではストリームの種別ごとに表 1 以下のような“ストリーム ID”を付与し、各ストリームを区別する。

各視聴ノードは、これらのストリーム ID のついたデータを流すそれぞれ独立した配信ツ

\*1 メッシュベースのストリーム分割については“Substream Trading” [5] を参照

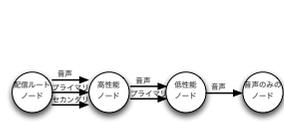


図3 ストリームを分割して必要なデータをリレー配信

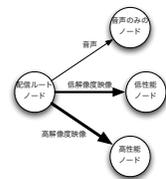


図4 ストリームを分割せず個別に配信



図5 ストリームを分割せず全てのデータをリレー配信

ストリーム ID	種別
0	音声
1	プライマリ映像
2	セカンダリ映像

表1 ストリーム ID

リーに参加し、必要なデータを取得する。\*1配信ツリー全体は、それら個別のツリーを重ねあわせたものになる。簡単のためストリーム ID が 0 および 1 のストリームのみを流すツリーの概略を図 6 に示す。

### 3.2.3 配信ツリーへの視聴ノードの参加

視聴ノードが映像の視聴を開始しようとした場合、以下の手順で配信ツリーに参加することになる。

- (1) 視聴したい映像をライブ送信している、または蓄積している配信ルートノードを探す
- (2) その配信ルートノードが形成する配信ツリーに参加する
  - 1 については 3.2.1 節で述べた。本節では 2 について述べる。

ある視聴ノードが既知の配信ルートノードが形成する配信ツリーに参加する場合、以下のような手続きにのっとって参加し、視聴を開始する。また、親ノードの離脱により受信するストリームが不足した場合も、同様の手続きでツリーに再参加する。

- (1) 予備ノード（後述）がある場合、その予備ノードに対して送信の親ノードになるように要求する。要求を受けた予備候補ノードは、即座に親ノードになる
  - (a) 予備ノードがなくなるか、または必要な全てのストリームについて親ノードが確保できるまで 1 を繰り返す
- (2) 視聴ノードは配信ルートノードを最初の親ノード候補請求先（候補請求ノード）とする

\*1 ただし、ストリーム ID1 のみ 2 のみといった受信はできない。かならず若い番号の ID から順番をとばすことなく受信する必要がある。つまり、0 のみ、0 と 1、0 と 1 と 2 が考えられる全ての組み合わせである。

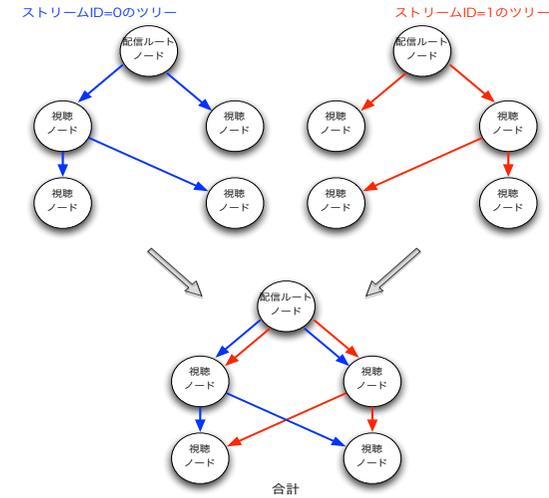


図6 ストリームごとのツリーを重ね合わせる

- (3) 視聴ノードが候補請求ノードに対し親ノード候補リストを請求する
- (4) 請求を受けた候補請求ノードは、以下のノードが含まれるリストを返す
  - その候補請求ノードからストリームを受信しているノード
  - その候補請求ノードにストリーム送信能力に余裕があれば、そのノード自身
- (5) 候補リストからノードを選択し、そのノードに対して送信の親ノードになるように要求する。要求を受けた親候補ノードは、自らのストリーム送信能力に余裕があれば、要求を受諾し親ノードになる
  - (a) リストから未要求のノードがなくなるか、または必要な全てのストリームについて親ノードを確保するまで 5 を繰り返す
  - (b) 全てのストリームの親ノードが確保できる前にリスト中の候補が尽きた場合、リスト中のどれかのノードを候補請求ノードとして 3 からやりなおす
- (6) 候補リストからノードを選択し、そのノードに対して予備ノードになるよう要求する。予備候補ノードは、自らのストリーム送信能力に余裕があれば、要求を受諾し予備ノードになる
  - (a) リストから未要求のノードがなくなるか、またはあらかじめ決められた数の予備ノードを確保するまで 6 を繰り返す
  - (b) 決められた数の予備ノードが確保できる前にリスト中の候補が尽きた場合、リスト中のどれかのノードを候補請求ノードとして 3 からやりなおす

実際にストリーム送信を受ける親ノードだけでなく予備ノードを確保しておくのは、親ノードの離脱から再参加までに要する時間を短縮するためである。

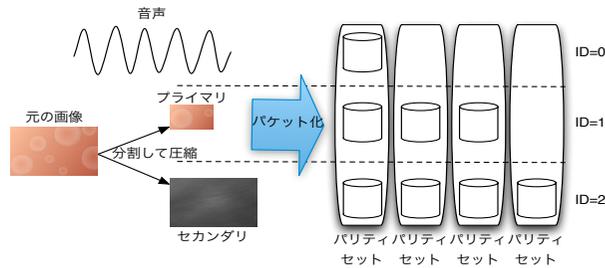


図7 映像データの RTP パケット化

パリティID	保護の対象とするデータストリームの内容
0	ストリーム ID 0 (音声)
1	ストリーム ID 0 および 1 (音声と低解像度画像)
2	ストリーム ID 0, 1, 2 (音声と高解像度画像)

表2 パリティID

### 3.2.4 配信ツリーからの視聴ノードの離脱

視聴ノードが視聴を終了する際には、以下の手順で参加しているツリーから離脱する。

- (1) 子ノードに対して離脱を通知する
- (2) 通知をうけた子ノードは代替となる親ノードを探し、ツリーに参加しなおす。その際新しい親ノードの探索方法は 3.2.3 節の手順による
- (3) ツリーに参加しなおした子ノードは、元の親ノード（離脱するノード）に対し繋ぎ直しの完了を通知する
- (4) 離脱するノードは、全ての子ノードの繋ぎ直しが完了するかタイムアウトを待ち、自らの親ノードに離脱を通知する
- (5) 離脱するノードは動作を終了する

### 3.3 品質の悪い経路での映像の訂正方法の実装

RFC 2733 [6] 方式をベースに、本システムで採用したツリー型配信システムに合わせたパリティ生成・利用による映像の訂正方法を開発・実装した。この節で紹介する方法を用いることで、不安定な P2P ネットワーク上で、ノードの離脱等が発生した場合でも、ストリーミング配信が途切れてしまうケースを減らすことができる。

#### 3.3.1 エラー訂正の単位

映像・音声データを実際に送信する際には、

- 分割され圧縮された映像データ
- 音声データ

をそれぞれ RTP [7] パケットに分解する。(図 7)

3.2.2 節で述べたように、音声、プライマリ映像、セカンダリ映像にはそれぞれストリーム ID 0, 1, 2 が割り当てられている。このときストリーム ID が重複しないようなパケットをセットにしたものをパリティセットと呼び、このパリティセットを単位としてエラー訂正のためのパリティパケット生成をおこなう。

#### 3.3.2 パリティストリームの生成

3.2.2 節で述べたように、本システムにおいては、各視聴ノードが受信する映像・音声データはそれぞれの視聴ノードごとに異なり、以下のようなパターンがありうる。

- ストリーム ID 0 のデータのみ受信する（音声のみ受信するノード）
- ストリーム ID 0 および 1 のデータを受信する（音声と低解像度画像を受信するノード）
- ストリーム ID 0 から 2 の全てを受信する（音声と高解像度画像を受信するノード）

これらの異なるパターンで組み合わされるデータストリームをエラー訂正で保護するためにパリティパケットを生成し、それらの種別ごとにパリティID を付与する。(表 2)

これらのパリティID を持ったパリティパケットは 3.3.1 節で述べたようにパリティパケットはパリティセットから生成されるが、このときストリーム ID  $x$  を持つ映像データパケットを  $s(x)$  とし、パリティID  $n$  を持つパリティパケット  $p(n)$  は以下ようになる：

$$p(n) = xor(s(0), s(1), \dots, s(n))$$

$$(n \geq 0)$$

この  $p(n)$  に対してヘッダ情報を付加したものが実際のパリティパケットになる。ただし、当該パリティセットに存在しないストリーム ID のパケットについては単純にスキップされる。つまり実際にパリティパケットの生成に使われるパケットの数は  $1 \sim n+1$  個の範囲の、パリティセットによって異なる数になる。仮に、あるパリティセットにストリーム ID  $m$  のパケットが存在しないとした場合、 $p(n)$  は以下ようになる：

$$p(n) = xor(s(0), s(1), \dots, s(m-1), s(m+1), \dots, s(n))$$

$$(n \geq 0, 0 \leq m \leq n)$$

こうして生成したパリティパケットを映像データのストリームとは別にストリーミング配信し、これをパリティストリームと呼ぶ。各視聴ノードは必要な映像・音声データのストリーミング配信ツリー (3.2.2 節) に加え、受信映像データの種類に応じたパリティストリームを受信することで、エラー訂正をおこなうことができる。ただし、パリティストリー

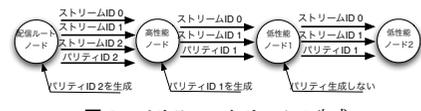


図8 パリティストリームの生成

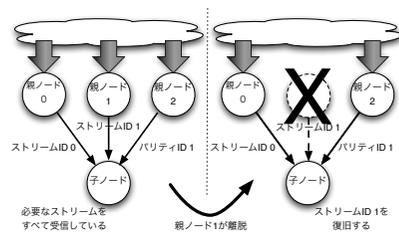


図9 親ノードの離脱により失われたパケットの再生成

ムの生成は必ずしも配信ルートノードではおこなわず、必要に応じて配信ツリー中の視聴ノードでおこなうこともある。

一般に各ノード（配信ルートノードおよび各視聴ノード）がパリティストリームを生成する条件は以下の通りである。

- 他のノードにパリティストリームの送信要求をされた
- 要求された ID のパリティストリームを受信していない

図8の「配信ルートノード」「高性能ノード」がこれにあたる。逆に「低性能ノード1」および「低性能ノード2」はパリティストリームの送信要求をうけていない（低性能ノード2）か、または要求された ID のパリティストリームを受信している（低性能ノード1）ため、自らはパリティストリームを生成しない。

これは不必要なパリティストリームの生成を避けるためである。

### 3.3.3 喪失したパケットの復旧

ツリー型の映像ストリーミングにおいて映像・音声データの packets が失われるのは以下のような場合である。

- ネットワーク障害。この場合障害が回復するか、障害が生じている経路を置き換えるまで packets の喪失が継続する
- 親ノードの不正な離脱。ストリームの送信元であるノードが通常の手順（3.2.4 節）によらずツリーから離脱すると、そのノードから送信をうけているはずの packets が喪失する。これは代替となる親ノードをみつけるまで継続する

いずれの場合においても、失われた packets が特定の映像ストリームひとつに属するものであれば、他のストリームで受信した映像・音声データの packets およびパリティ packets から復旧することができる。（図9）このとき復旧される packets は、 $r$  を失われた packets が属するストリームのストリーム ID、 $s(x)$  をストリーム ID $x$  の映像データ packets、 $n$  を

そのノードが受信しているパリティストリームのパリティ ID、 $p(n)$  をパリティ packets としたときに、以下の式であらわされる：

$$s(n) = xor(s(0), s(1), \dots, s(r-1), s(r+1), \dots, s(n), p(n))$$

$$(n \geq 0, 0 \leq r \leq n)$$

ただし、各 packets は同一のパリティセットに含まれているとする

## 4. まとめ

著者らは、能力の低い携帯端末を含む、さまざまな能力を持つノード向けの映像配信システムを実現するために、複数の解像度での再生が可能な形式に映像を圧縮および変換する技術、および変換された映像を蓄積し、P2P ネットワーク上で構成したツリー構造を用いて配信する技術を開発し、分割された映像データおよび音声データについてそれぞれツリーを構成し、配信するシステムを開発・実装した。

今後は、アルゴリズムの改良やユーザインターフェースの実装、さらに実証実験を行う予定である。

## 参考文献

- 1) 安倍広多. P2P システム上での安定したサービス提供基盤 musasabi. 情報処理学会研究報告, Vol. 2009-IOT-4, pp. 131-136, 2009.
- 2) 秋山大輔, 細川和宏, 安倍広多, 石橋勇人, 松浦敏雄. Z 曲線を用いた効率的な 2 次元位置情報の分散管理手法の提案とその評価. 情報処理学会研究報告, Vol. 2009-IOT-8, No.9, 2010.
- 3) FMJ Project. Freedom for media in java. 入手先 <http://fmj-sf.net/>. (参照 2010-05-24).
- 4) 吉田幹, 奥田剛, 寺西裕一, 春本要, 下條真司. マルチオーバーレイと分散エージェントの機構を統合した P2P プラットフォーム PIAX. 情報処理学会論文誌, Vol.49, No.1, pp. 402-413, 2008.
- 5) Zhengye Liu, et al. Substream trading: Towards an open p2p live streaming system. In *The 16th IEEE International Conference on Network Protocols*, pp. 94-103, 2008.
- 6) J.Rosenberg and H.Schulzrinne. An RTP Payload Format for Generic Forward Error Correction. RFC 2733 (Proposed Standard), December 1999. Obsoleted by RFC 5109.
- 7) H. Schulzrinne, S.Casner, R.Frederick, and V.Jacobson. RTP: A Transport Protocol for Real-Time Applications. RFC 3550 (Standard), July 2003. Updated by RFCs 5506, 5761.