

M-021

# オーバレイマルチキャストを利用した複数チャンネル配信方式の検討

## A Study of Multiple Channel Delivery Methods with Overlay Multicast

大谷 晋一郎 †   謝 夢 ‡   綾木 良太 ‡   島田 秀輝 †   佐藤 健哉 ‡  
Shinichiro Ohtani   Xie Meng   Ryota Ayaki   Hideki Shimada   Kenya Sato

### 1 はじめに

現在、スポーツ中継やビデオ会議など、ライブ環境におけるストリーミング配信の分野においてオーバレイマルチキャストが利用されている。オーバレイマルチキャストは、インターネットの上位層に構築されたオーバレイネットワーク上でマルチキャストを行う技術であり、ネットワークポロジに影響されないマルチキャストを実現できる。オーバレイマルチキャストにより、サーバの負荷や管理コストの軽減が実現され、高画質データの配信が可能となった。しかし、一方で配信の際の遅延や通信帯域の効率的な利用、複数のチャンネルが存在する際のスムーズなチャンネル切り換えなどの問題がある。

本稿では、DHT(分散ハッシュテーブル)を用いることで複数チャンネルの配信を実現したオーバレイマルチキャストのネットワークにおいて、これらの問題を解決できる手法を提案し、その配信方式の検討を行う。

### 2 関連研究

#### 2.1 Scribe

Scribe[1]はDHTのアルゴリズムであるPastry[2]を用いたオーバレイマルチキャストのインフラストラクチャである。Pastryは探索対象となるノードのIDと自ノードのIDとのプレフィックス長が最長となる近隣ノードを選択し、中継する。この処理を繰り返し行うことによって最終的に目標ノードへと到達することができる。Pastryの探索方法を図1に示す。

Scribeでは図2に示すように、Pastryが作成した探索経路をマルチキャストにおける配信木として利用する。各ノードはjoinメッセージを配信元のノードまで送信することでマルチキャストに参加することができる。ScribeはPastry上で実装されるため、Pastryの特性であるスケーラブルなネットワークと信頼性を兼ね備えたマルチキャストを実現することができる。

#### 2.2 Scribeの問題点

ネットワークに参加しているノードは配信木の上位ノードからデータを受け取り、そのデータを自ノードの子ノードの数だけ複製し、送信することでマルチキャストを実現する。各ノードが持つことができる子ノードの最大数は各ノードの上り帯域幅によって決まる。ネットワークに参加している各ノードの上り帯域幅はそれぞれ異なるため、各ノードが持つことができる子ノードの最大数はノードによって異なる。Scribeではこのノード毎の子ノードの最大数の違いについて想定されていない。従って、配信元に近い上位層に子ノードの最大数が小さいノードが配置されると配信木が横方向に広がらず、必

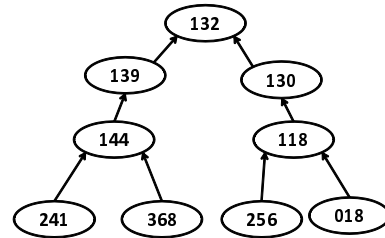


図1 Pastryの探索手法

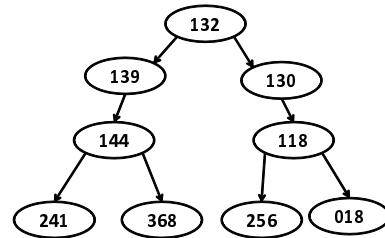


図2 Scribeの配信木

要以上に長い配信木が構築される。例えば、図2ではIDが‘130’と‘139’のノードが上位層に配置されている。しかし、これらのノードは子ノードを1つしか持つことができない。もし、下位層に配置されているノードがより多くの子ノードを持つことができるならば、そのノードを上位層に配置することでより低い配信木を構築することができる。

### 3 提案手法

#### 3.1 前提条件

DHTを用いることで、複数チャンネルを配信できるオーバレイマルチキャストのネットワークを実現する。利用環境としてスポーツ等のライブ動画配信を想定し、ライブ動画は競技場から各利用者のPCまで配信されるものとする。各チャンネルは1台のカメラに対応しており、利用者はチャンネルを切り替えることによって、好きなカメラからのライブ動画を視聴することができる。ネットワーク上の各チャンネルとノードには固有のIDが割り当てられる。ノードはDHTを用いてチャンネルのIDを探索することで各チャンネルに接続できる。DHTのアルゴリズムにはネットワークでの近傍性が考慮されているPastryを用い、マルチキャストにはPastry上でのマルチキャストを実現できるScribeを用いる。

#### 3.2 概要

Scribeはノード毎の帯域幅が異なる場合、必要以上に高い配信木を構築してしまう可能性があるという問題が

† 同志社大学 理工学部 情報システムデザイン学科

‡ 同志社大学大学院 工学研究科 情報工学専攻

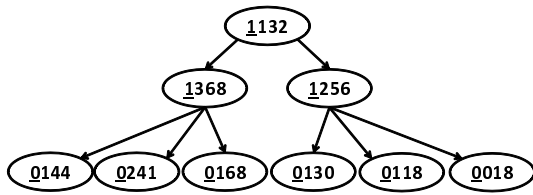


図3 提案手法の配信木

あった。本稿では、ネットワーク中の各チャネルを含めたノードの ID の前に評価値を付加することによって、子ノードを多く持つことができるノードを配信木の根に近い上位層に配置し、従来の Scribe より低い配信木の構築を実現する。これ以降、ネットワーク中の他のノードより広い帯域幅を持ち、多くの子ノードを持つことができるノードを高性能ノードと呼ぶ。

配信木の根となる各チャネルと高性能ノードには評価値 '1' を、それ以外のノードには '0' を割り当てる。Pastry の探索では、チャネルの ID と各ノードとのプレフィックス長を用いるため、最上位ビットに配置された評価値が各チャネルと異なるノードはプレフィックス長が '0' となり、チャネルからの探索経路上の距離が遠い位置に配置される。Scribe では、配信木に Pastry が作成した探索経路を利用するため、評価値が異なるノードは配信木では下位層に配置される。従って、各チャネルと評価値が等しい高性能ノードは相対的に上位層に配置される。この手法を用いることによって、図 2 のような配信木においても下位層に高性能ノードがあれば、図 3 に示すように従来の Scribe より低い配信木を構築することができる。図中で下線が付けられた数字が評価値である。

## 4 評価

### 4.1 評価条件

Scribe と提案手法の配信木構築のアルゴリズムに対してプログラミングによる評価を行う。マルチキャストに参加するノード数による配信木の高さの変化と参加ノード全体に含まれる高性能ノードの割合による配信木の高さの変化をそれぞれ従来の Scribe と比較し、提案手法の性能を評価する。ノード数に対する評価では、ネットワークへの参加ノード数を 0~10000 まで増やすことで、参加ノード数がそれぞれの手法で構築された配信木に与える影響を評価する。高性能ノードの割合に対する評価では 1000 ノードが参加しているネットワークにおいて、ノード全体における高性能ノードの数を増加させることで、高性能ノードの割合が配信木に与える影響を評価する。各評価において、通常のノードは 2 つの子ノードを、高性能ノードは 5 つの子ノードを持つことができると仮定する。

### 4.2 結果と考察

ノード数と高性能ノードの割合の変化に対する評価結果を図 4 と図 5 にそれぞれ示す。ノード数に対する評価からは提案手法のほうが従来の Scribe の手法より低い配信木を構築することができていることがわかる。また、高性能ノードの割合に対する評価からは、ネットワーク中の高性能ノードの割合が低い条件において提案

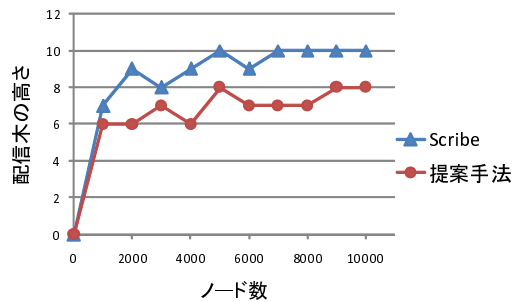


図4 ノード数に対する評価

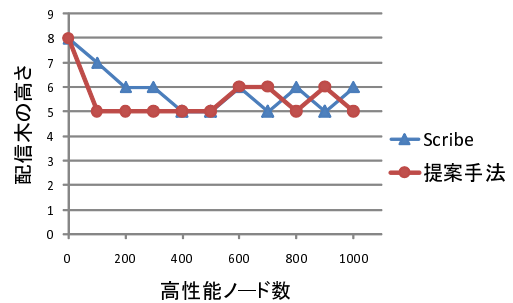


図5 高性能ノードの割合に対する評価

手法のほうが低い配信木を構築できていることがわかる。これらのことから、多数のノードが参加する大規模な配信システムや多数の高性能ノードの参加が期待できないようなシステムでは、提案手法のほうが適していることがわかる。

## 5 まとめ

本稿では、オーバーレイマルチキャストを利用した複数チャネルの配信を実現するために、DHT の利用を提案した。DHT を利用したマルチキャスト手法には既に Scribe が提案されているが、各ノードの帯域幅を考慮していなかったため、ノードの有効利用や配信木の効率的な構築という点で問題があった。しかし、帯域幅を考慮した評価値を ID に付加することによって従来の手法より低い配信木を構築することができた。低い配信木の構築は各ノードの帯域幅を有効利用できるだけでなく、各ノードと配信元との経路が短くなるため、配信の際の遅延の減少やスムーズなチャネル切り替えの実現にも効果が期待できる。

## 参考文献

- [1] Castro, M., Druschel, P., Kermarrec, A.-M. and Rowstron, A.: SCRIBE: A large-scale and decentralized application-level multicast infrastructure, *IEEE Journal on Selected Areas in Communications*, Vol. 20, No. 8, pp. 100-110 (2002).
- [2] Rowstron, A. and Drusche, P.: Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems, *IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, Vol. 2218, pp. 329-350 (2001).