

オーバレイネットワークにおける 複数の予備経路を利用した経路再構築手法の検討

高木 健士[†] 北 望[†] 重野 寛[‡]
慶應義塾大学大学院理工学研究科[†] 慶應義塾大学理工学部[‡]

オーバレイネットワーク上での大容量データの配信技術として、アプリケーションレイヤマルチキャスト (ALM) が注目を集めている。ALM には、マルチキャストをネットワークインフラに変更を加えることなく運用できるという利点がある。しかし同時に、しばしば接続が不安定になるという問題があり、ノード離脱後のネットワークの再構築が大きな課題となっている。本稿では、ALM における、複数の予備経路を利用した経路再構築手法を提案する。提案手法では、各ノードが親の離脱に備え、予め予備経路を複数確保することで、配送ツリーの再構築後も配送ツリー長を短く保ち、データ転送遅延を抑えることが可能である。また、シミュレーションによる比較評価により、配送ツリー長を短く保つことが示された。

A Study of Multi-Proactive Route Maintenance for Overlay Network

Kenji TAKAGI[†] Nozomu KITA[†] Hiroshi SHIGENO[‡]
Faculty of Science and Technology, Keio University[†]
Graduate School of Science and Technology, Keio University[‡]

Application Layer Multicast(ALM) has been attracting attention as a delivery technology for large-capacity data on the Overlay Network. In ALM, we can manage multicast without changing network infrastructure. However, there is often a problem that connection becomes unstable, and reconstruction of a network after a secession of parent the multicast session becomes a big problem. In this paper, we propose a multi-proactive route maintenance for ALM. In our proposed algorithm, each node prepares for the secession of parent and finds proactive routes. Thus, we can keep a length of delivery tree short after reconstruction and suppress a data transmission delay. Computer simulations are carried out and it is shown that our proposed algorithm achieves better performance compared to existing method.

1 はじめに

ADSL や FTTH などのブロードバンドの普及に伴い、動画や音声などのマルチメディアコンテンツへの需要が増加してきた。特に動画は、大容量サイズのものが多く、配信サーバやネットワークにかかる負荷が懸念される。例えば、ライブストリーミングなどは受信者の数が多くなることが予想されるので、コンテンツ配信方法は重要な課題である。これに対して、アプリケーションレイヤマルチキャスト (ALM) が注目を集めている。ALM には、マルチキャスト機能をエンドノードに持たせることにより、既存のインフラを変更することなく運用することが

できるという利点がある。

しかしその一方で、いくつかの問題点もある。最も大きなものは、中継ノードがユーザ端末であるために接続が不安定になることである。例えばあるノードが、マルチキャストグループから離脱すると、そのノードが中継している全てのノードがデータを受信できなくなるという問題が生じる。

このとき、データを受信できなくなったノードはマルチキャスト配送ツリーの再構築を試みるのだが、ツリー再構築の手法としてはいくつかのアプローチがある。そのうちの1つとして、上位ノード離脱後の再接続先を予め探索しておき、予備経路として利

用する手法が挙げられる。本研究では、この予備経路を利用する手法の問題点を改善し、予備経路の作成手法及びそれに基づいた配送ツリーの再構築手法を提案する。

予備経路作成における既存手法 [1] は、ツリーに冗長性を持たせることで確実に再接続することができ、ツリー再構築の時間を短縮することに成功している。しかしこの手法には、再構築後にツリーが深くなってしまい、データ転送遅延が大きくなるという問題点がある。そこで本提案方式では、ツリー再構築のための予備経路を複数作成する。具体的には、既存手法の予備経路を1本確保した後、さらに複数の予備経路を作成する。そして、帯域に余裕がある場合に限り、複数の予備経路を利用して配送ツリーを再構築する。これにより、既存手法より再構築後の配送ツリー長は短縮し、データ転送遅延を小さくすることが可能となる。

2 ALM 構築

ALM はノード間でコネクションをリレーさせることで、擬似的なマルチキャストを実現する。このとき、各端末ノードは IP 層を考慮する必要はないため、ネットワーク構築にはアプリケーション層の親を考えればよい。

ALM では各ノードは親ノードからデータを受信し、子ノードにデータを配信する。また、ある時点で受け付けることのできる子の数を degree と言う。その最大値を max-degree と言う。

ALM は用途に応じていくつかのタイプがあるが、本稿では単一ソースノードによるツリー型の ALM を対象とする。これはツリーのルートにソースノードが位置するもので、ライブストリーミングなどの放送型アプリケーションに適している。

オーバーレイネットワーク構築アルゴリズムはノードが、join, leave, failure 時に用いられる3つのサブアルゴリズムにより構成される。本稿では特にツリー型の ALM である Peercast を例に挙げ、各イベント発生時のオーバーレイ構築アルゴリズムを説明する。

2.1 join アルゴリズム

新規ノードがセッションに参入するとき、新規ノードはまずソースノードへ接続要求を送信する。ここでソースノードは帯域に余裕があるのであれば新規ノードを受け入れる。そうでなければ現在送信しているいくつかのノードの中から1つを選択し、そのノードへリダイレクトさせる。この手順を繰り返す

ことにより新規ノードはいずれかのノードと接続し、データを受信する。

2.2 leave アルゴリズム

セッション参加中のノードが離脱するとき、離脱ノードはまず親と全ての子に離脱メッセージを送信する。メッセージを受信した親はそのノードとのセッションを開放する。子は親ノードの消失によりそのままではデータを受信できなくなるため、新しい親を見つける必要がある。ここでは離脱ノードの親、すなわち祖父にあたるノードへ再接続要求を送信する。祖父の情報は離脱メッセージにより与えられる。

2.3 failure アルゴリズム

ALM は中継ノードが信頼性の低い端末ノードであるため、一度張ったオーバーレイリンクが確立されているかどうかを定期的に確認する必要がある。各ノードは親と子に対して一定間隔でメッセージを送信する。これをハートビートメッセージといい、一定時間経過してもメッセージが届かない場合は障害が発生したと判断してこのリンクを切断する。その際の対応は leave アルゴリズムとほぼ同様であるが、故障ホストの子は祖父にあたるノードを知らないのソースノードへ再接続要求を送信する。

3 関連研究

いかなる ALM も 2.1~2.3 に示したイベントが発生した場合にオーバーレイ構築が行われる。特に leave, failure 発生時にはセッション参加中のノードが親を更新しなくてはならない。多くの ALM は親の離脱を知ってから新しい親を探し始める。これでは親が見つかるまでに時間を要する場合があり、最悪バッファがアンダーフローすることも考えられる。

これを回避するために、予め親候補、すなわち予備の経路を定めておくことによりこの時間を短縮する手法が提案されている [3][1]。

[3] は受け持つ子供のみならずデータを配信するのではなく、ごく一部のデータをツリー上の離れたノードに配信し、これを予備経路とする。そのノードの上位ノード障害が発生した場合、一部だけでなく全てのデータを送信する。しかし、この方法は経路の管理をデータパケット自体を使って行うため、一部しか流さないとしてもオーバーヘッドは非常に大きい。さ

らに予備経路はランダムに選択されるため、すべてのノードが確実に予備経路を保持するわけではない。

[4] は予備経路の作成に当たり、コントロールパケットを用いており、新規ノードがツリーに参入する度に予備経路を計算する。新規ノードの親ノードが離脱する仮定の下、その親子間で全域木を作ること为目标としている。しかし、(1) 式を満たしていない場合、着目ノードの親子間だけでは全域木を作ることができない。(1) 式において、 d は未使用の degree 数、 c は子、 n は子供の数を表す。この時、着目ノードより下位に位置するノードを用いて左辺を増やすことになるので、スケーラビリティに問題がある。ノードの参加・離脱の度に、その親と祖父にあたるノードは (1) 式を計算をする必要があり、それによってコントロールオーバーヘッドも増加する。

$$\sum_{j=0}^{n-1} d(c_j) \geq n-1 \quad (1)$$

そして、本研究の基礎となる研究 [1] では、[4] の問題点を解決するため、各ノードは degree が 1 以上になるようにオーバレイツリーを構築する。そして、新規ノードがツリーに接続すると、親ノードの離脱を仮定して予備経路の構築を行う。以下に [1] における予備経路の作成手順を説明する。

1. まず新規参入ノードの親ノードは自分の親（新規参入ノードの祖父）に子の情報を送信する。
2. これを受信した祖父は全ての孫ノード ping パケットを送信する。
3. 最も早く応答した孫ノードは祖父からの経路を予備とする。
4. その次に応答した孫ノードはその前に応答したノードからの経路を予備とする。

(以降孫ノードが全て予備経路を保持するまで 4 の繰り返し)

1~4 より、図 1 のように予備経路が構築される。これにより各ノードは参加・離脱の度に (1) 式を計算する必要がないので、子の degree 情報を保持しなくてもよく、コントロールオーバーヘッドを抑えることができる。また、各ノードが最低 1 つ degree を余らせていることから、どのノードに着目しても (1) 式を必ず満たしており、親子間で確実に予備経路を構築することができる。

しかし実際には、離脱や障害により予備経路への切り替えが行われた際に degree の余りが 0 になる

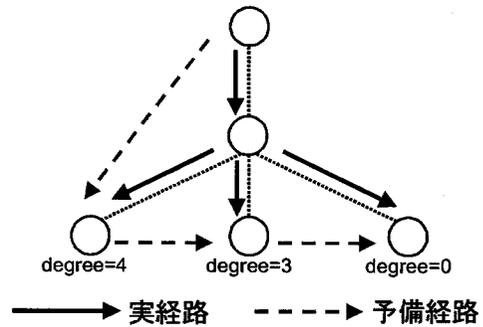


図 1: 予備経路作成の例

ノードが生じる可能性がある。この場合、そのようなノードは自分の子にメッセージを送信し、degree が 2 以上余っているノードを探す。メッセージを受信したノードは、degree が 2 以上あれば最後に受け付けたノードを自らにリダイレクションさせ、なければさらに自分の子に転送する。これにより全てのノードが degree を 1 つ余らせる状態を保持する。

この手法は、新規ノードがツリーに接続された後に、その周辺ノードのみで確実に予備経路を作ることができるので、既存のほぼ全てのツリー型 ALM に対応することができる。

この手法の問題点としては、degree を 1 つ余らせ予備経路に使うため、degree が 1 以下のノードをどう扱うかということが挙げられる。degree が 0 のノードは全く配信ができないので、[1] だけでなく、全ての ALM で共通の問題である。このようなノードは、ツリーの“葉”に位置する他に仕方がない。また、[1] はツリーに冗長性を持たせているため、degree が 1 のノードに関しても同じ問題がある。予備経路を構築する際、先に述べたように [1] は (1) 式を必ず満たすことになる。しかし、degree が 1 以下のノードが集まる場合、(1) 式を満たせなくなる可能性がある。

そこで、各ノードは degree が 1 以下のノードを 1 つだけしか子として受けれない、という条件をつける。そして、図 1 に示すように、degree が 1 以下のノードを強制的に予備経路の末端に配置するようにすれば、この問題を回避できる。

[1] では、ALM において低オーバーヘッドかつ離脱ノードの親子間で確実に予備経路を構築することが可能となっている。

4 提案アルゴリズム

4.1 複数の予備経路作成

関連研究 [1] では、子ノード多く保持するノードが離脱した場合、配送ツリー再構築後のツリー長が長くなるという問題がある。ツリー長が長くなってしまうと、データ遅延が発生してしまう。そこで本研究では、再構築後もデータ遅延が発生しないように予備経路を複数作成し、ツリー長を短く保つ手法を提案する。予備経路の作成に当たり、まずは関連研究 [1] と同様の手法を使用する。本提案では1本の予備経路を作成した後、更に複数の予備経路を作成する。

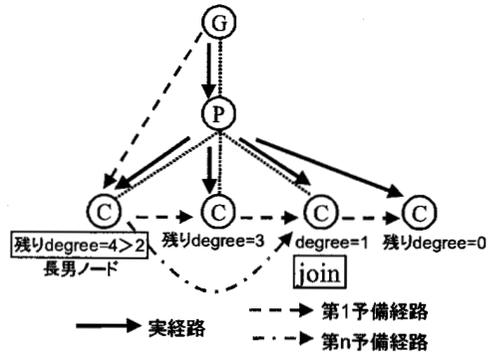


図 2: 第 n 予備経路の作成

4.2 予備経路の種類

本提案において、1つのノードから作成される予備経路は、2つの種類に分類することができる。

- 第1予備経路
親ノード離脱時に、次の接続先として保証される経路である。本提案においても、各ノードは関連研究 [1] 同様ストリーム1本分の帯域、つまり degree を1以上に保っておき、この再接続先として保証されている経路のことを第1予備経路と定義する。
- 第 n 予備経路 ($n: 2$ 以上の自然数)
親ノード離脱時の接続先として保証されていない経路である。本提案の中心的役割を担う経路であり、親ノード離脱後の再接続先としては保証されておらず、degree に余裕がある場合に限り接続できる経路である。親ノード離脱時にこの経路を利用することで、配送ツリー長を短く保つことが可能である

4.3 第1予備経路の作成

本提案における予備経路作成アルゴリズムでは、新規ノードが参加した時点で、まず第1予備経路を作成する。第1予備経路作成手順は以下の通りである。

1. 新規参加ノードの親ノードは、自分の親(新規ノードの祖父)に子供の degree 情報を送信する。
2. 子供の中で、残り degree が最大のノードは、祖父からの経路を予備経路とする。このノードを長男ノードとする。
3. その次に大きな degree を持つノードは、長男ノードからの経路を予備とする。
4. さらに次に大きな degree を持つノードは、1つ前のノードからの経路を予備とする。

5. 以降子ノードが全て第1予備経路を持つまで繰り返す。

1~5より、図のように予備経路が作成される。この操作を、新規ノードが参加するたびに行う。

4.4 第 n 予備経路の作成

4.4.1 第 n 予備経路作成手順

第1予備経路の作成に続いて、第 n 予備経路を作成する。新規ノードが参加した場合の、第 n 予備経路の作成手順は以下の通りである。図2にその様子を示す。ここで、Gは祖父ノード、Pは親ノード、Cは子ノードを示す。C1が長男ノード、C3を新規ノードとする。

1. 新規ノードが参加した時点で、第1予備経路を作成する。
2. 長男ノードの残り degree が3以上であれば、長男ノードから新規ノードへの経路を作りこれを第 n 予備経路とする (n はその時点で保持している予備経路の本数で決定する)。

4.4.2 第 n 予備経路が作成されない場合

第 n 予備経路は、以下の場合作成されない。

- 長男ノードの degree が2以下の場合
親ノードが離脱した時、ノードの degree が0になってしまうのを防ぐため、第 n 予備経路は適用されない場合がある。そのため本提案では、新規ノードが参加した段階でノードの degree が2以下の時、第 n 予備経路を作成しない。
- 新規ノードが長男ノードになった場合
第 n 予備経路は新規ノードと長男ノードの間

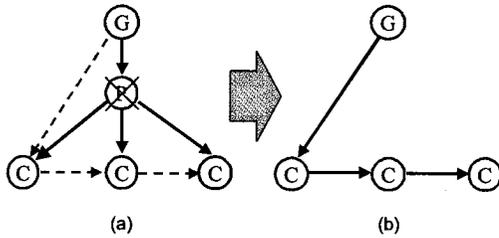


図 3: 関連研究による再構築手法

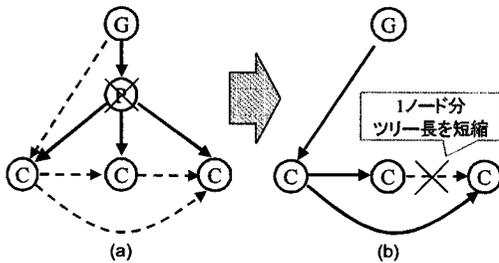


図 4: 複数の予備経路による再構築手法

で作成されるため、新規ノードが長男ノードになった場合は第 n 予備経路を作成しない。

4.2 で述べたように、第 n 予備経路は仮に作成されたとしても、再構築時において長男ノードのネットワーク帯域に余裕がない場合は適用されない。こうすることで、第 n 予備経路の追加によってネットワーク帯域を圧迫することがなくなる。また、第 1 予備経路は新規ノードが参加する度に更新されるが、第 n 予備経路は新規参加ノードと長男ノードの間で作成され、離脱するまでその情報を保持し続ける。

4.5 経路再構築手法

ノード離脱後、離脱ノードの子が保持している予備経路情報に基づいて配送ツリーの再構築が行われる。関連研究 [1] による手法でツリーの再構築を行うと、図 3(b) のように、再構築後のツリー長が長くなる。

本研究における予備経路を複数利用した手法では、図 4(b) のように再構築後でもツリー長を短く保つ。図の例では、[1] と比べて 1 ノード分ツリー長を短縮できている。また、ここで仮に第 n 予備経路が適用されなくとも、第 1 予備経路は確保できているため、再接続先を失うことはない。

表 1: 実験で使用するパラメータ

ノード数	20, 30, ~, 100
1 ノードあたりの平均 d_{degcc}	3, 6, ~, 10
参加・離脱回数	計 1000 回

5 シミュレーション評価

5.1 シミュレーション概要

提案アルゴリズムの性能評価のため、シミュレーションを行い、以下の項目に関して評価した。

- 配送ツリー平均長
- 配送ツリー再構築時間

比較対象として、関連研究 [1] による既存手法のみの場合と、提案手法を加えた場合の 2 種類を用いた。シミュレーションで用いるパラメータの値を表 1 に示す。

ノード数と 1 ノードあたりの平均 degree を組み合わせて行った。以下、実験の手順を説明する。

1. 表 1 に示したノード数を配送ツリーに参加させる。
2. ノードの参加・離脱を 1 : 1 の割合で計 1000 回繰り返す。
3. 2 を 1 セットとし、100 セット実行する。

5.2 シミュレーション結果

5.2.1 配送ツリー平均長

配送ツリー平均長に関するシミュレーション結果を図 5 に示す。ここで、平均長は、式 (2) のように求める。

$$(\text{配送ツリー平均長}) = \frac{\sum_{j=0}^{n-1} E(j)}{n} \quad (2)$$

n は join イベント、leave イベントの総回数を示し、 $E(j)$ は各イベント時の配送ツリーの全体の深さである。

図 5 からわかるように、平均 degree の値が小さいときは既存手法と提案手法に差は見られない。これは degree が小さければ子供を多くつなげることが出来ず、第 n 予備経路が適用されにくいからである。

一方、平均 degree の値が大きくなるにつれ、既存手法の配送ツリー平均長が深くなっているのがわかる。これは既存手法の予備経路作成手法では、ノードが子供を多く保持している分だけ離脱時にツリー長が深くなってしまったためである。

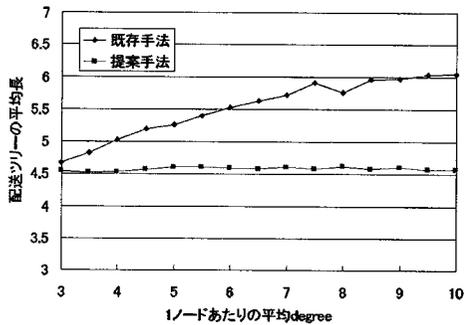


図 5: 配送ツリー平均長

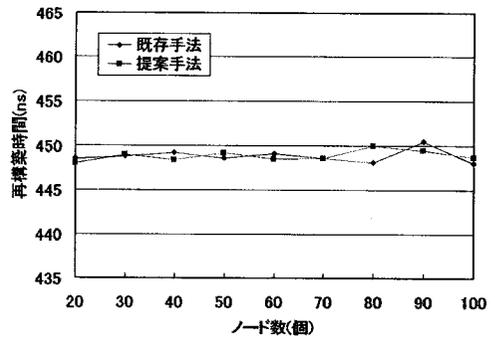


図 6: 配送ツリー再構築時間

また提案手法では、degreeが増えてもツリーの平均長はほぼ一定であることがわかる。これは、以下の2点のツリーが浅くなる要素と深くなる要素があったためである。

1. 浅くなる要素

1 ノードあたりの degree が大きくなると新規ノードのツリー参加位置が浅い位置になり、全体的にツリー長は短くなる。

2. 深くなる要素

提案手法は帯域に余裕がある場合に利用する経路であるため、ノードが子を多く保持してかつ帯域にも余裕がなければ、離脱時に配送ツリーが深くなり易い。

5.2.2 配送ツリー再構築時間

シミュレーション上で、ノードが離脱してからツリーを再構築するのに要した時間を図6に示す。

図6からわかるように、既存手法と提案手法の間に差はなく、再構築時間も一定を保っている。これは提案手法により予備経路を複数作成することに対する負荷はないことを示している。また、再構築は子・親・祖父の3世代のノードの間で行っているため、ノード数が増えることで再構築時間が増えることはなく、スケーラビリティが高いと言える。

6 おわりに

ALMにおいて、ノード離脱時の配送ツリーの再構築は大きな問題である。既存の手法[1]では、再構築後にツリー長が長くなるという問題があった。そこで本稿では、複数の予備経路を利用した経路再構築手法を提案した。この手法は、各ノードが親の離脱

に備え、予め予備経路を複数確保することで、再構築後のツリー長を短く保つことを実現する。シミュレーション評価を通じて、本手法を用いた場合、既存手法に比べて再構築後のツリー長が短くなることを示した。また、再構築にかかる時間は本手法と既存手法で差異がなかった。これにより、本手法は既存手法に比べて高い性能と実用性があることが示された。

参考文献

[1] T. Kusumoto, J. Katto, and S. Okubo. Proactive route maintenance for tree-based application layer multicast and its implementations. *IEICE TRANS, INF. SYST.*, 2006.

[2] S. Deshpande, M. Bawa, and H. Garcia-Molina. Streaming live media over peers. *Technical Report 2002-21*, Vol. 21, , Mar. 2002.

[3] S. Banerjee, S. Lee, B. Bhattacharjee, and A. Srinivasan. Resilient multicast using overlays. *In Proceedings of ACM SIGMETRICS 2003*, June. 2003.

[4] M. Yang and Z. Fei. A proactive approach to reconstructing overlay multicast trees. *In Proceedings of IEEE INFOCOM 2004*, 2004.