

ALMにおける高速再接続のための親ノード探索法の提案と評価

The proposal and evaluation of parent node search method for rapid reconnection in ALM

篠田 智治† 中島 潤‡ 広奥 暢‡
Tomoharu Shinoda Jun Nakajima Tohru Hirohku

1. はじめに

近年、インターネット上で音声や動画のデータファイルを公開するサイトが發展し、以前に比べマルチメディアコンテンツ作成者は手軽に自作のコンテンツをユーザに配信可能になった。しかし、これらのストリーミング配信方法はオンデマンド方式であり、リアルタイムマルチメディアコンテンツを配信することは出来ない。一方、ライブストリーミング方式での配信を行うためには、ユーザからの同時大量アクセスに耐える配信用サーバと、広帯域回線が必要であり、個人で用意するのは難しいと言える。

この問題を解決する方法として ALM (Application Level Multicast) が注目されている。ALM では、メンバ管理、パケットの複製、パケットの転送はすべてオーバーレイ・ネットワークに仮想的に接続されたノードがユニキャストで行う。この技術により個人でも手軽にライブストリーミング配信を行うことが出来る。しかし、ALM には中継ノードがオーバーレイ・ネットワークから離脱すると、そのノードが中継していた下流のノードすべてが切断され受信不能な時間が発生してしまう。この時間を断続時間と言う。

本研究では、ライブストリーミング配信の断続時間減少を目的とした高速再接続のための親ノード探索法を提案し、その有効性を検討する。

2. ALM (Application Level Multicast) と課題

ALM は、P2P 技術をライブストリーミング配信に適用した方式である。コンテンツ配信元は、一部のノードにのみ直接配信を行い、他のノードは配信元から直接配信を受けたノードからバケツリレー方式(図1参照)で配信を行う方式である。それゆえ、同時視聴者数が増加しても配信元からの配信トラフィックは一定に保つことができ、配信者への負担が少ない。

中継ノードがオーバーレイ・ネットワークから離脱すると、そのノードが中継していた下流のノードすべてに断続時間が発生してしまう。特に配信元に近い上流ノード

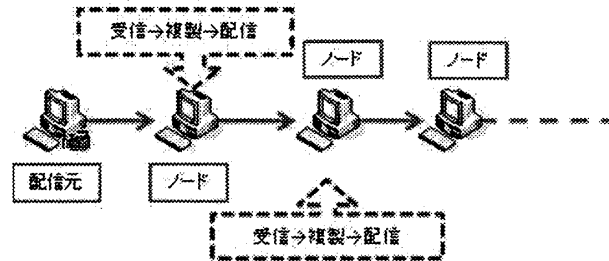


図1 ALMの配信方式

ドが離脱するとネットワークに与える影響は大きいと言える。それゆえ、すべてのノードに対し断続時間を発生しないように、また発生したとしても実現可能な限り微小な時間にする必要がある。

これらを解決する方法として冗長木を利用した予備親保持などが挙げられる[3]。各ノードが予備親を保持することにより、自分の上流のノードが離脱した際でも、効率の悪い再接続を行うことなく、上流ノードを切り替えることにより断続時間を縮減することが出来る。このように、なるべく効率の悪い再接続を実行しないよう耐障害性を高める研究がなされている。しかし、耐障害性をどれだけ高めたとしても不可避の障害は必ず存在する。その際、現状のままでは断続時間が発生してしまう。

3. 高速再接続のための親ノード探索法の提案

本研究では不可避の障害が発生した場合に、現状の再接続手順よりも断続時間を縮減する高速再接続の手法を提案する。

ALM には大別してツリー型とメッシュ型があり、ツリー型は単一ソースノードによる動画配信などに用いられ、メッシュ型はテレビ会議などに用いられる。本稿では、ツリー型の ALM を対象とする。

ツリー型ストリーミング配信用 ALM の代表として Gnutella プロトコルを使用した PeerCast[4]が挙げられる。この Gnutella プロトコルの再接続における課題は、新規接続と再接続の手順が等しく、参加までの時間がかかってしまうことである。その主な原因として Gnutella は PureP2P モデルであるため、どのノードもトポロジー全体を把握していないことが挙げられる。そこで、再接続の高速化のために配信元ノードでトポロジー全体の把握を行い、最適なノードにフォワーディングを行うように改

† 北海道情報大学大学院
Graduate School of Hokkaido Information University

‡ 北海道情報大学 情報メディア学科
Faculty of Information Media,
Hokkaido Information University

良する。これは以下の Gnutella パケット[5]の拡張により解決する。

①最下層のノードが上位のノードから ping パケットを受信したら pong パケットに自分の識別子を乗せて返答する, ②識別子を受け取ったノードは自分の識別子を追加し, 上位のノードから来た ping パケットに対し pong パケットで上位ノードへ通達する, ③以下, 配信元ノードまで繰り返す, ④配信元ノードは受け取った pong パケットより経路表を作成する。

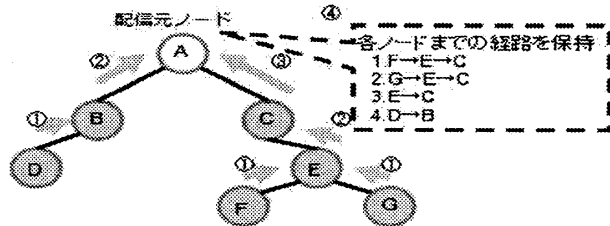


図2 Gnutella パケットの拡張

これにより, 配信元によるトポロジー制御が可能となり, ループが発生しないように再接続先ノードを探索する。

4. 提案の有効性の検討

提案の有効性を示すために, 提案手法で示したトポロジー制御を用いた手法と, 従来の手法の ping パケットの送信回数を比較した。前提条件として提案手法, 従来手法共に再接続の最短 ping パケット送信回数は配信元に接続する 1 回になることから, 今回は最長の場合のみ比較することにする。

ノードの数を N と設定し, ノード間の遅延・帯域を等しく設定した ALM ツリーと仮定する。一定時間後トポロジーが定まったら接続可能なノードを M , 最大階層を L とおき, N' と ping パケットが一致したときに再接続完了とする。また, 新規追加ノードは接続不可能とする。配信元以外のノードが離脱した場合を考え, 再接続が必要な状況の手順を C ライクな言語で以下に示す。

I Gnutella の再接続

```
i = 0; //ping パケットの送信回数
j = 0; //フラグ
k = 0; //最大階層のスタック
while(j < 1){
    if(i == M){
        break;
    }
    else if(i == L){
        i = i - L;
        k = k + 1;
    }
    i++;
}
```

```
}
k = k * L;
i = i + k;
```

II 提案手法の再接続

```
a = 0; //ping パケットの送信回数
b = 0; //フラグ
while(b < 1){
    if(a == M){
        break;
    }
    else if(i == L){
        break;
    }
    a++;
}
```

このときの a が ping パケットの送信回数になる。

I では i が最下層の L 以上に ping パケットを送る可能性があるのに対し, II では L 回目でループから抜け出す。よって $a \leq i$ が成り立つので提案手法では断続時間が短くなると言える。

5. まとめ

本研究では, アプリケーションレベルマルチキャストにおける断続時間の縮減を目的とした高速再接続のための親ノード探索法の提案を行った。提案手法では, 配信元がトポロジーを制御することによって, Gnutella の再接続でロスと考えるループの時間を短縮することが出来た。

今後の課題として, 今回は設定しなかったノード間の遅延や帯域, さらにノードの数などの環境を想定し, それぞれ重み係数を付け, シミュレーション検証を行う必要がある。

これらを加味して断続時間の短縮のためにさらに効率の良い高速再接続のための親ノード探索法を模索したい。

参考文献

- [1] 江崎浩, P2P 教科書 “インプレス R&D”, 2008
- [2] 篠田智治, 中島潤, 広奥暢, “PeerCast 向け高速再接続機能の提案”, 電子情報通信学会, 2008
- [3] 國近洋平, 甲藤二郎, 大久保榮, “予備親探索機能を有したアプリケーションレベルマルチキャスト”, 信学技法, IN2003-210, 2004
- [4] PeerCast, <http://www.peercast.org/>
- [5] Gnutella Protocol Development, <http://rfc-gnutella.sourceforge.net/index.html>