

L-011

大規模コンテンツ配信のためのアプリケーションレベル・マルチキャストツリー構築に関する検討
A Study on Application-Level Multicast-Tree Construction for Large Scale Content Distribution清水 直人[†]亀山 渉[†]渡辺 裕[†]

Naoto SHIMIZU Wataru KAMEYAMA Hiroshi WATANABE

1 はじめに

大規模なコンテンツ配信を実現するためのメカニズムとして、近年アプリケーションレベル・マルチキャスト (ALM) が盛んに研究されている [1][2][3][4][5]。ALM では、各エンドホスト (以下、ホストと呼ぶ) でパケットの複製、マルチキャストルーティングなどのマルチキャストに関する機能を実現する。すなわち、ALM では、IP マルチキャストルータではなく、各ホストがマルチキャストツリー構築の役割を担う。これにより、IP マルチキャストのようなネットワークインフラストラクチャの変更を要求することなく、多受信者に対して効率の良いコンテンツ配信が実現できる。我々は、Motion JPEG2000[6] のような多階層を有するデータのストリーミングについても検討を行っており、そのような場合においてもきちんとデータ間の同期の確保を行えば、ALM は有効なソリューションであると考えている。しかし、その一方で、IP マルチキャストルータと比べ、セッションへの参加、離脱などが頻繁に起こる安定性のないホストがマルチキャストツリーの構築を行うことになる。したがって、ALM において、ホストの参加、離脱などのネットワーク状況の変化に追従できるマルチキャストツリー構築手法が必要不可欠である。本稿では、ALM に適したマルチキャストツリー構築手法についての基礎検討を行った。

2 ALM ツリーの構築

ALM では、セッションに参加している各ホストを以下の 2 つのトポロジに組織化する [7]。

1. コントロールトポロジ

予期せぬホストの離脱などを発見し、マルチキャストツリーの再構成を行うために必要なメッセージを定期的を送信するためのトポロジ。

2. データトポロジ

実際のコンテンツデータを配信するためのトポロジ。

ALM では、基本的にコントロールトポロジをメッシュ型で構築し、データトポロジをツリー型で構築する。ツリー型で構築されるデータトポロジが ALM ツリーである (以下、データトポロジを ALM ツリーと呼ぶ)。コントロールトポロジをメッシュ型で構築する理由として、1) 早期に離脱ホストの発見、ALM ツリーの再構成が行える、2) 交換されるメッセージはコンテンツデータに比べ、サイズが基本的に小さく、メッシュ型にしても帯域を圧迫しない、ことが挙げられる。ALM は、コントロールトポロジ、ALM ツリーの構築方法により、以下の 2 つのアプローチに大別することができる。

1. Mesh-first approach

ホスト間でメッシュ型のコントロールトポロジを構築し、それを基に ALM ツリーを生成する [1]。

2. Tree-first approach

ホスト間で ALM ツリーを構築し、その後、メッシュ型のコントロールトポロジを構築する。コントロールトポロジ構築の仕方として、1) ランダムにホストを選択して構築する方法 [2][3]、2) 隣接ノード間でクラスタを構成し、それを基に構築する方法 [4][5]、がある。

ALM ツリー自体の構築は、IP マルチキャストの場合と同様に、以下に示すプロセスにより行われると考えられる [9]。

1. 初期ツリーの構築
2. ツリーの部分的な改良 (tree rearrangement)
3. 新しい最適ツリーへの移行 (tree migration)

3 検討課題

[4][5] では、上述したとおり、隣接ノード間でクラスタを構成し、コントロールトポロジを構築している。クラスタ単位の局所的な情報により ALM ツリーを維持するので、ホストの参加、離脱などのネットワーク状況の変化に追従することが可能になる。他の ALM と比べて、大規模コンテンツ配信に適していることがシミュレーションにより示されている。[4][5] では、naive multicast routing[8] を用いて初期 ALM ツリーを構築する。その後、ALM 木の部分的な改良 (tree rearrangement) をクラスタリングにより構築されたコントロールトポロジを基に行う。しかし、以下の検討課題について議論されていない。

検討課題 1 ホストの頻繁な参加、離脱により引き起こされる ALM ツリー的大幅な品質劣化に対応するための ALM ツリー再構成 (tree migration) への対応

検討課題 2 上記 ALM ツリー構築手法の妥当性

検討課題 3 クラスタのサイズ

ALM ツリー構築手法の妥当性とは、最適ツリーにどの程度近い ALM ツリーを構築できるかということである。

3.1 クラスタリングを用いた最小全域木構築

そこで、本稿では、クラスタリングを用いた最適ツリー構築手法について提案し、上記で挙げた検討課題についても議論する。最適ツリーは、コンテンツ配信ホストが 1 台のみの場合は、最短木、複数台の場合は最小全域木が妥当である。本稿では、最小全域木を最適ツリーとする。手法の概要図を、図 1 に示す。

クラスタ内のホストのみ局所的な情報を用いて最小全域木を構築するので、ツリーを構成するホストの状態が変化した場合においても、その変化に追従可能な最小全域木構築が可能となる。これにより、ホストの変化に追従するという ALM の特性にマッチした ALM 最適ツリーの構築が可能になる (検討課題 1)。

しかし、一方で、局所的な情報のみを用いて、最小全域木を構築していくため、全てのホストを用いて最小全域木を構築した場合と比較して、準最適な最小全域木となるはずである。両者の性能差については、シミュレーションを通じて明らかにする (検討課題 2)。

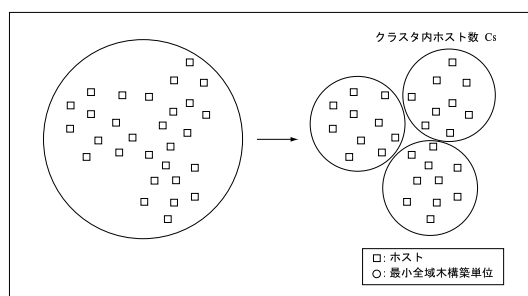


図 1 クラスタリングを用いた最小全域木構築

[†]早稲田大学大学院 国際情報通信研究科

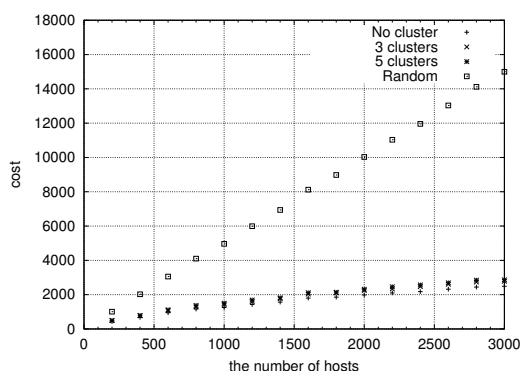


図2 ホスト数とツリーコストの関係

また、クラスタサイズについて、以下のように定式化できると考えられる(検討課題3)。

$$C_s = f(\text{Control_traffic}, \text{Computation_cost})$$

Control_trafficは、ホスト間の帯域を占めるコントロールメッセージの比率、Computation_timeは、最小全域木構築に要する計算コストである。具体的な関数 f については、今後の課題である。

4 シミュレーション

4.1 評価手法

シミュレーションは、次のように行った。まず Waxman法 [10] を用い、ノード数 5000 のフラットなランダムトポロジーを構築した。その際、トポロジジェネレータとして BRITE [11] を用いた。次に、5000 ノードの中から、ランダムに N ノード選択し、それをホストとした。したがって、残り $5000 - N$ ノードは、ルータとなる。そして、1) ホスト全体で最小全域木を構築した場合、2) クラスタを単位として最小全域木を構築した場合、3) ホスト間をランダムに接続する全域木を構築した場合(項目1のみ)、について以下に示す2項目の測定を行った。

項目1 ツリーコスト

項目2 ツリー構築までに要する時間

項目1は、クラスタを単位として最小全域木を構築した場合に、ホスト全体を考慮に入れて算出する最適な最小全域木と比べてどの程度コストが劣るのかを、項目2は、計算コストがどの程度削減されているか、についてそれぞれ表している。計算コストの削減は、変化により素早く追従できるツリーを構築できるということである。

最小全域木は、ホスト間の遅延をコストメトリックとして、プリムのアルゴリズム [12] を用いて構築した。クラスタを単位としての最小全域木の構築は、クラスタ内最小全域木とクラスタ中心同士の最小全域木を組み合わせることで行った。クラスタを単位として最小全域木を構築した場合において、クラスタ構築時間はツリー構築時間には含まない。(実行環境 - CPU: Pentium III 700MHz, RAM: 256MB, OS: Windows XP, Java 環境: JDK1.4.0)

4.2 評価結果

図2にホスト数とツリーコストの関係を示す。図3にホスト数とツリー構築時間の関係を示す。

図2より、クラスタを単位として構築した最小全域木は、ホスト全体で構築した最小全域木と比較して、ほとんど性能差がないことが分かる。また、図3よりホスト全体で最小全域木を構築する場合と比較して、より短時間で最小全域木を構築していることが分かる。すなわち、クラスタリングを用いて、最小全域木を構築することにより、ある程度の性能を持つ最小全域木をより素早く構築することが可能になることが分かる。

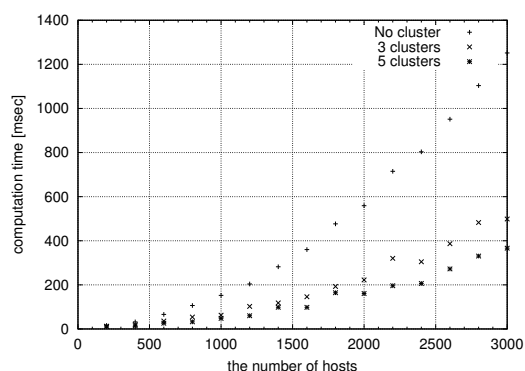


図3 ホスト数とツリー構築時間の関係

5 まとめ

本稿では、クラスタリングを用いた最適ツリー構築手法を提案した。シミュレーションにより、クラスタリングを用いて最小全域木を構築しても、ホスト全体で構築した最小全域木と比べ、ほとんど性能差がないことを示した。また、ホスト全体での最小全域木の構築時間と比較し、大幅に削減可能であることも示した。これにより、ALMのように、ホストの状態が頻繁に変化するような場合において、最適ツリーを構築、維持していくための手法として本手法が有効であることが分かる。

謝辞

この研究は、TAO 委託研究課題“通信ネットワーク利用放送技術の研究開発”のサポートによる。

参考文献

- [1] Y. Chu, S. Rao, and H. Zhang, “A case for End System Multicast,” ACM SIGMETRICS, June 2000.
- [2] P. Francis, “Yoid: Extending the Multicast Internet Architecture,” <http://www.aciri.org/yoid/>
- [3] B. Zhang, S. Jamin, and L. Zhang, “Host multicast: A framework for delivering multicast to end users,” IEEE INFOCOM’02, June 2002.
- [4] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, “Scalable application layer multicast,” ACM Sigcomm, Aug. 2002.
- [5] D. A. Tran, K. A. Hua, and T. T. Do, “A Peer-to-Peer Architecture for Media Streaming,” To appear in IEEE JSAC Special Issue on Advances in Overlay Networks, 2003.
- [6] Takahiro Fukuhara, David Singer, “Motion JPEG 2000 Final Committee Draft Version 1.0,” ISO/IEC JTC 1/SC 29/WG 1, N2117, Mar. 2001.
- [7] S. Banerjee, and B. Bhattacharjee, “Comparative Study of Application Layer Multicast Protocols,” Under submission
- [8] Doar and Ian Leslie, “How Bad is Naive Multicast Routing,” IEEE INFOCOM’93, Apr. 1993.
- [9] A. Chakrabarti and G. Manimaran, “A Case for Scalable Multicast Tree Migration,” IEEE GLOBECOM’01, Vol. 3, pp.2026-2030, Nov. 2001.
- [10] B. Waxman, “Routing of Multipoint Connections,” IEEE J. Select. Areas Commun., Dec. 1988.
- [11] “BRITE,” <http://www.cs.bu.edu/brite/>
- [12] Robert Prim, “Shortest connection networks and some generalizations,” Bell System Technical Journal, vol.36, pp.1389-1401, Nov. 1957.