データ駆動型ALMによるデータ配信手法について

越賀 雅士† 太田 義勝† 鈴木 秀智†
†三重大学大学院工学研究科情報工学専攻

1 はじめに

近年では，プロセッサの高速化，メモリの大容量化，ネットワークの広帯域化を背景として，多くのユーザが高画質の映像をインターネットを通して視聴できるようになった．しかし，従来のクライアント/サーバ型では，サーバへの負荷が集中することになり，大規模配信を行うには大規模サーバを必要としていた．これに対して，近年P2P型での映像配信が注目されている．この映像配信の手法をアプリケーション層マルチキャスト (Application Layer Multicast，以下ALM) という．

本研究では，ALMの転送方式の1つデータ駆動型に注目し，データ駆動型ALMで制御パケット量を抑えつつ，ノードの上り帯域を効率良く使用することによって遅延を少なくする手法を提案する．

2 ALMの転送方式

ALMの転送方式としてツリー型，メッシュ型，そしてデータ駆動型の3つが挙げられる．

2.1 ツリー型

ツリー型は，ノードがマルチキャストのためのデータ配信木を作成し，木構造に沿って根から葉に向かってデータを配信する．利点として，データ配信経路の管理や操作，最適化を行いやすい点がある．欠点として，ノードの離脱などが起こると素早く修復する必要があり，もし修復に問題があればデータ配信に失敗することがある．既存手法としてSplitStream[1]などが挙げられる．

2.2 メッシュ型

メッシュ型は，ツリー型が持つ親子関係の制約を無くし，複数の隣接ノードへfloodingでデータを転送する．データの受信が初めてであれば，受信したノードを除いた全ての隣接ノードに対してデータを転送する．受信済みであれば転送しない．利点として，他の隣接ノードからデータ受信ができるため，ノードの離脱に対して素早く修復する必要がない．欠点として，無駄なデータ転送が多いので，それを低減する工夫する必要がある．既存手法としてBullet[2]などが挙げられる．

2.3 データ駆動型

データ駆動型は，メッシュ型のトポロジを構成する．データを隣接ノードへ一方的に送るのではなく，隣接ノードからの要求を受けて送信する方式である．利点として，メッシュ型で起こる無駄なデータ転送を無くすことができる．欠点として，要求を受けてから，データを送信するのでツリー型やメッシュ型に比べると遅延が大きくなる．また，データ保持情報などの制御パケットの通知頻度でも遅延の大小は変わってくる．

3 関連研究

データ駆動型ALMの関連研究として，CoolStreaming[4]やChainsaw[5]が挙げられる．どちらの場合もデータ転送方法として，データを分割してセグメントに分け，セグメント毎に送受信を行う．セグメント保持の有無については，BitMap(以下BM)を用いる．配信ノードのBMは全て1となる．以下ではCoolStreaming，Chainsawでのデータ転送方式と問題点について述べる．

3.1 CoolStreaming

ストリーミングデータを1000msec毎に分割し，セグメント毎で送受信を行う．各ノードは隣接ノードと一定時間毎にBMの交換を行う．この手法の問題点として，BMの通知頻度が高いと制御パケット量が増大してしまい帯域の無駄使いをしてしまうことが挙げられる．また，BMの通知頻度が低いと新たなセグメントを取得しても隣接ノードへの通知が遅れてしまい，隣接ノードのセグメント取得の時間が遅くなってしまうことが挙げられる．

3.2 Chainsaw

データを16KB毎に分割し，セグメント毎で送受信を行う．各ノードは新たなセグメントを取得した時に，隣接ノードへBMの通知のための制御パケットを送信する．この手法の問題点として，セグメント受信毎にBMを送信することになるので，制御パケット量が増大してしまい帯域の無駄使いをしてしまうことが挙げられる．

A Method for Data Delivery on Data-Driven ALM
†Masashi Koshika †Yoshikatsu Ohta †Hidetomo Suzuki
†Division of Information Engineering, Graduate School of Engineering, Mie University

4 提案方式

CoolStreaming では一定時間毎に BM の送受信を行うことにより，BM 転送回数は一定数に抑えることができるが，新たなセグメントを取得してもすぐには隣接ノードへ通知されず，セグメントの送受信が一定期間されない場合がある．つまり，上り帯域の使用をうまく活用できていない問題がある．また，Chainsaw のように新たなセグメントを取得した時に BM の送受信を行う時では、上り帯域の使用効率は問題ないが，BM 送受信量が多くなってしまい，通信処理の負担が大幅に増えてしまう．これに対して，データ転送時における BM の送信を全ての隣接ノードへセグメントを送信し終えたときに行うことで，各ノードの上り帯域を効率良く使用し，BM の送信量を抑え，遅延を少なくできると考えられる．

4.1 提案方式におけるデータ配信の流れ

1. 各ノードは隣接ノードへ BM を送信する
2. 隣接ノードは受け取った BM を解析し，保持していないセグメントを要求する
3. 隣接ノードからの要求を受けたノードはセグメントを各隣接ノードへ送信する
4. 全ての隣接ノードへセグメントを送信したら，再び隣接ノードへ BM を送信する

後は２～４を繰り返し行うことでデータ送受信を行う．また各ノードは上記のデータ転送を行っている時に，同時に隣接ノードからセグメントの受信を行っている．

4.2 評価方法

今回提案するシステムの評価値として，上り帯域使用率 (Outbound Bandwidth Rate of Use，以下 OBRU)，平均 BM 送信回数 (Average Number of BM Transmission, 以下 ANBT)，全体遅延時間 (Total Delay Time，以下 TDT) の３つを用いる．OBRU は，式 (1) で求めれる．

$$OBRU = \frac{\sum S}{T} \times 100 \quad (1)$$

ここで，S は各セグメントの送信時間．T は全体時間である．ANBT は，式 (2) で求められる．

$$ANBT = \frac{\sum B}{N} \quad (2)$$

ここで，B は各ノードの BM 送信回数．N は参加ノード数である．

TDT とは，データ転送開始時間から全ての参加ノードがデータを受信するまでの時間である．

シュミレーション実験で以上の評価値を求める．

5 実験

本実験は，計算機上に各ノードをスレッドで動かした仮想ネットワーク上で行った．各ノードの隣接ノード数を 3～10 と変化させて行った．転送データサイズは 5MB で，16KB のセグメントに分割してデータ転送を行った．BM 送信サイズは 128bit である．ネットワーク参加ノード計 100 ノードを順次参加させていく．参加にはデータ配信ノードであるソースノードへ接続し，他の参加ノードの情報を取得する．全てのノードが参加し，通信状態を確立したらデータ転送を始める．参加している全ノードがデータを受信したらシステムを終了する．以上の実験を計 5 回行った．

実験結果を表 1 に示す．

表 1: 実験結果

評価値	min	average	max
OBRU(%)	23.24	29.73	34.97
ANBT(回)	13.56	17.24	22.45
TDT(ms)	27452	31492	38573

OBRU についてはデータ転送を開始してから平均で約 30 %ほど上り帯域を使用している状態となった．ANBT については平均で約 17 回となり，TDT においては平均で 31492ms となった．

6 おわりに

データ駆動型 ALM で新たに制御パケットを抑えつつ，上り帯域を効率良く使用して，遅延を少なくする手法を提案した．今後の課題として，インターネット環境を考慮したシミュレーション実験を行うことにより，既存手法との比較実験を行う予定である．

参考文献

[1] M. Castro, et al, “SplitStream: High-bandwidth Content Distribution in a Cooperative Environment”, IPTPS, 2003.

[2] D. Kostić, et al, “Bullet:High Bandwidth Data Dissemination Using an Overlay Mesh”, SOSP, 2003.

[3] 池長慶彦，首藤一幸，村岡洋一，“実ネットワークに適応するオーバレイマルチキャスト放送基盤”，情報処理学会研究報告 2007-DSM-44, 2007.

[4] X. Zhang, et al, “CoolStreaming/DONet: A Data-Driven Overlay Network for Peer-to-Peer Live Media Streaming”, IEEE/INFOCOM, 2005.

[5] V. Pai, et al, “Chainsaw:Eliminating Trees from Overlay Multicast”, IPTPS, 2005.