

P2P ストリーム放送のためのルーティング実装とその性能評価

2005MT101 柴田 美沙登
指導教員

2005MT124 渡邊 藍子
後藤 邦夫

1 はじめに

現在,コンピュータの性能向上,ブロードバンドネットワークの普及によりデジタルコンテンツ技術が注目されている.そして,P2P(Peer to Peer)技術の発達により,誰でも配信者となるP2Pストリーミング(PeerCast, ShareCastなど)が実現された.P2Pストリーミングでは,各ノードがストリームデータを受信すると同時にデータを複製し,下流のノードにデータを受け渡す方法を用いる.しかし,P2Pストリーミングでは,ストリームデータを中継しているノードが接続を中断した場合,その下流ノードへのストリーミングが中断されてしまう.そして,再接続するための時間がかかり,その間ストリームデータが受け取れない問題がある.

本研究は,過去の卒論(発行2008年)[1]の引き継ぎである.[1]では,ストリームデータを中断しているノードが接続を中断した場合に,うまく再接続し,ストリーミングを再開することができない.また,エミュレーション実験での評価が記述されているが,実際にどの様に行われたのか分からず,本当に改善されているのかが分からない.よって本研究では,ノードが接続を中断した場合に効率良く再接続し,ストリーミングを再開できるように,再実装する.また,ネットワークエミュレータであるGoto's IP Network Emulator(以下GINE)[3]を使って模倣し,複数のノードで動かして動作テストを行い,改善前後の評価を比較することも行う.

岩瀬,山田の卒論(発行2009年予定)は,過去の卒論(発行2006年)[2]とは別のルーティング提案,シミュレーションを行っている.

柴田は主にプログラム作成を担当,渡邊は主に実験環境構築,評価を担当した.

2 従来のP2Pとルーティングについて

P2Pとは,クライアント・サーバ型とは対照的に参加するコンピュータが相互に接続され,ファイルなどの情報を直接通信しやりとりする技術のことである.

2.1 既存のルーティング方式

既存のルーティング方式では,ノード間の距離を考慮したルーティング方式と接続を考慮したルーティング方式がある.

- ・ノード間の距離を考慮したルーティング方式

iphop数によるノード間の距離を計算し,一番近いノードと接続する.距離が近いのでデータの転送速度は早くなる.しかし接続が切れた場合,再度接続先を探して接続しなおすため,時間を要してしまう問題がある.このルーティング方式をA方式とする.

- ・ノード間の接続を考慮したルーティング方式

各ノードが他のノードとの接続を維持するかをストリーミング参加時に自己申告させる.

- ・常時参加ノード...ストリーミング終了まで接続を維持する

- ・一時参加ノード...ストリーミング終了まで接続を維持しない

常時参加ノードには,複数の下流ノードが接続される.一方一時参加ノードには途中で離脱する可能性があるため下流ノードは1つだけとなる.しかしこの方式では,一時参加ノードの下に常時参加ノードが接続されることがある.そのとき一時参加ノードが離脱した場合,下流の常時参加ノードから下の全てのノードが中断されてしまう.また距離は考慮されないため,遠くのノードと接続され,データの転送速度が遅くなってしまいう問題がある.

3 提案されたルーティング方式

A方式の問題を解決するため,過去卒論[2]で考案された提案ルーティング方式を説明する.このルーティング方式をB方式とする.

ネットワークへのトラフィック軽減,データの品質の劣化を防ぐため,ノード決定方法が提案された.そのノード決定方法に従ってノードを決定する.そのノード決定方法は3.1節で説明する.

P2Pストリーミングでは,ノードが離脱した場合,データ通信が中断され再び接続するため時間がかかってしまうという問題がある.これを回避するため,接続先ノードとは別にもう1つノードを決めておく.そうすることによって,接続先のノードが離脱した場合,あらかじめ決めておいた別のノードにすぐにつなぎ直すことによって,再接続に要する時間の軽減が期待できる.図1にシステムの流れを示す.

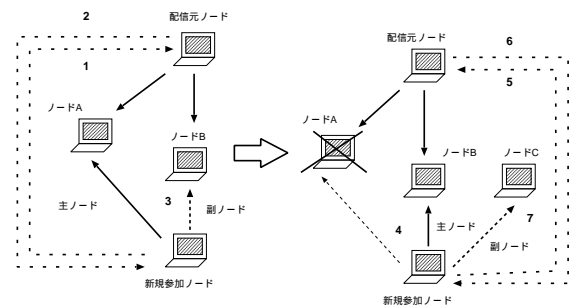


図1 提案されたルーティングモデル(B方式)

番号1~7は図1に準ずる.

番号1~3は,ストリームデータを受け取るまでの流

れを示す。

番号 4~7 は、接続先のノードが離脱したときの再接続までの流れを示す。

1. 配信元ノードは、参加者ノードの情報を持っている。新規参加ノードは、欲しい情報を配信元ノードに問い合わせをする。
2. 参加者ノードの情報を送る。
3. 送られてきたノード情報をもとにノード決定方法に従ってノードを決定し、一番優れているノード A を主ノードとして接続し、次に優れているノード B を副ノードとして確保しておく。そして主ノードからストリームデータを受信する。
4. 主ノードとしていたノード A からの配送が切れた場合、主ノードがなくなりデータを受信できなくなってしまうので、副ノードで確保しておいたノード B を主ノードに切替える。
5. 副ノードを決定するため、配信元ノードに問い合わせる。
6. 配信元ノードはノードの情報を送る。
7. 送られたノード情報をもとにノード決定方法にしたがってノードを新しく計算し、一番優れているノード C を副ノードとして確保する。

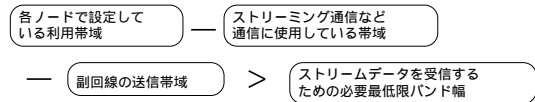
3.1 ノード決定方法

ノードを決定するのは全ての参加者ノードからではなく、候補ノードとして限定されたノードから決定する。候補ノードに必要な情報を以下に示す。

- P2P 中継回数 ($p2phop$)：配信元からのストリームデータの的中継回数。各ノードでストリームデータを複製し、下流ノードへ渡しているのので、この中継回数が増えるとデータの品質が劣化してしまう。
- IP 中継回数 ($iphop$)：新規参加ノードと配信元ノードとの $iphop$ 数。この数が少ないとネットワーク全体のトラフィックを減らすことができる。
- 初期設定バンド幅 ($bands$)kbps：各ノードが利用可能なバンド幅。ストリームデータを送信するバンド幅のみの計算。これを設定することによりノードへの負担を防ぐ。
- 必要バンド幅 ($bandw$)kbps：ストリームデータを受信するための必要最低限のバンド幅。
- 毎秒ごとに送るパケット ($mainp$)：ノードが各下位ノードに送るパケット。
- 副ノードの数 ($scou$)：あるノードに対して経路を確保している下位ノードの数。数を設定すれば、1つのノードに負荷が集中することを防ぐ。

上記の情報から候補ノードを決定する式を示す。式にすると

$$bands - mainp - bandw \times scou > bandw \quad (1)$$



この (1) 式を満たすノードが上流ノードを決定する条件となる。これは利用可能なバンド幅より必要バンド幅が大きいとストリームデータを送信できないので、その問題を防ぐためである。

次に (1) 式を満たす時

$$x = \alpha \times p2phop + \beta \times iphop +$$

$$\gamma \times \frac{1}{bands - mainp - bandw \times scou} \quad (2)$$

$$\alpha + \beta + \gamma = 1 (\alpha > 0, \beta > 0, \gamma > 0)$$

この (2) 式で、3つの値を重みつき平均を行い足すことによって得られる x の値が一番小さいものから順に主ノード、副ノードが決定される。

4 システム改良

1. `setupRoute` で主ノード、副ノードを見つけ出す操作が効率が悪いと考えたので、見つけ出す方法を変更した。
2. P2P 通信を行っている間、一定時間ごとにノードが離脱しているかしていないか確認して、離脱している場合、離脱したノードの IP アドレスを知らせてくれるようにプログラムを追加した。このようにすることにより、データ送信先ノードが突然離脱したときに、過去の卒論 [2] で提案されたようすぐにノードを切替えることができる。この処理を `isAlive` というメソッドで行った。
3. 主ノード、副ノードにしているノードが離脱した場合、再度 3.1 節のノード決定方法で計算しなおし、主ノード、副ノードを新たに決定し、再接続するシステムを追加した。この処理を `change` というメソッドで行った。

4.1 システム改良 (`setupRoute`)

`setupRoute` は、ノードを計算し、主ノード・副ノードを決定するメソッドである。

ノード計算に必要なノード情報をあらかじめ入手しておく。各ノードにおける残りの利用帯域が、ストリームデータを受信するための必要最低限バンド幅より高いとき通信を開始できるので、この条件の元で、3.1 節で説明したノード決定方法にしたがって、ノードを計算する。計算結果の値から一番小さいものを主ノード、次に小さいものを副ノードとする。値が同じ場合は、ノード番号の小さいノードが優先される。主ノード、副ノードが決まったらその番号とノード情報を経路表に書き込み、通信を開始する。

4.2 システム改良 (isAlive)

isAlive は、接続しているノードが離脱したかどうかを確認してくれるメソッドである。

接続しているノードに ping 要求を出し、その要求に対して返事が返ってきたら、「切れてません」と同時にその IP アドレス、ポート番号を表示させる。返事が返ってこない場合は、離脱したとみなし、「切れました」と同時にその IP アドレスとポート番号も表示させる。またここでは、1 回の ping 要求で返事がなく、離脱したと判断してしまうと、ping 要求の返事が返る前に、他のノードからの ping 要求またはコマンド応答を受け取ってしまい、離脱していないのに離脱したと判断してしまうので、3 回送って 3 回とも返って来なかったらノード離脱と判断している。そして、主ノード、副ノードとしていたどちらかのノードが離脱したと判断した場合、新たに受信するノードを決定しなければいけないので、“chang”メソッドで処理を行う。

4.3 システム追加 (change)

離脱時に、ノード切替えの動作を行うメソッドである。このメソッドは以下の動作をする。

- 離脱したノードを主ノードとして使っていた場合、副ノードを主ノードにつなぎ換えて副ノードを新たに見付けて指す。
- 離脱したノードを副ノードとして使っていた場合、副ノードを新たに見付けて指す。

新たなノード決定方法は、setupRoute メソッドのノード決定方法と同じ方法で行う。

5 今後の改良点

ノード切替えがうまくいかず、プログラムが不完全である。

chang メソッドで、新しくノードを捜し出す際、周辺のノード情報がうまく入手できず、新しいノードを決定するためのノード計算ができない。原因としては、離脱したときに配送先リストが変更されていない点がある。離脱したノードのアドレスなどの情報をリストから消し、全く離脱したノードの情報がない状態で、新たにノードを見付けなければならない。その動作ができていないので、うまくいかないのだと考える。したがって、今後システムを完成させるには、配送先リストを変更するメソッドを新たに追加する必要がある。

6 実験

OS は Linux4.2, カーネルは 2.6.20 を使用した。

6.1 ネットワークモデル

我々が考案するシステムは、100 名程度が利用する小規模なマルチキャストシステムと想定している。しかし、実際に全ての P2P ユーザを動作させて実験を行うことが困難なため、また GINE では、遅延やパケットロスなどを比較的簡単に、自由に設定できるため、GINE を使ってエミュレート実験をすることにした。さらに

UML(User Mode Linux)ではなく、仮想ネットワークスタックを使用する理由は、仮想ネットワークスタックはネットワークの部分だけを仮想的に作るためである。また、カーネルを少し変更しなければならないという欠点があるが、UML を使用するよりも効率が良いと考えたためである。

図 2 の様に、ネットワークを仮想的に作成した。Queue で遅延を発生させ、IX(Internet eXchange)を再現している。また、NS0~NS2 はそれぞれルータを再現し、NS3~NS12 の NS10 個はそれぞれ PC のネットワークの部分再現している。

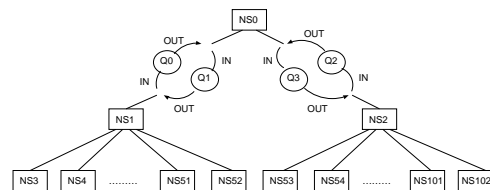


図 2 仮想ネットワーク (ノード 100 個)

6.2 expect スクリプト

仮想ネットワーク中で、P2P プログラムを実行時に expect スクリプトを利用する。expect コマンドは、対話的なプログラムのやりとりを自由化するプログラムである。プログラム独自のコマンドを仮想ネットワークを作成するためのファイルに書き込んでも、上手く実行できないため、expect コマンドを利用した。P2P プログラム実行時、ノードごとに利用 IP アドレス、ポート番号が違っているので、ノードごとに expect ファイルを用意した。

1 つのコマンドが終了する前に、次々とコマンドが実行されてしまう。P2P プログラムが、起動する時間軸通りに動くように、sleep コマンド、および『expect “~” = “~”』と出力されたら』という処理でコマンドを調整した。

6.3 実験結果

過去の卒論 [2] で B 方式を利用して実装したものと、2.1 節で紹介した距離に注目した A 方式とを GINE を使いエミュレーションし、P2P プログラムを expect スクリプトで動かし、ファイルに動作状況を書き込んで接続状態を確認し、データを取り、比較して評価する。

過去の卒論 [1] では、ノード 10, ノード 100 で P2P プログラムが正常に動くかどうかは確認されていない。今回我々は、ノード 10 の仮想ネットワークと、ノード 100 の仮想ネットワークで、P2P プログラムを起動した。ノード 10 では確実に P2P プログラムを正常に動かすことができたが、ノード 100 ではノード 20 までしか P2P プログラムを正常に動かすことができなかった。しかし、2 台の PC を使い、手作業ではノード 100 まで P2P プログラムを起動できることを確認した。また、PC 自体のメモリは十分に存在したことも確認した。このことから、仮想ネットワークでノード 20 までしか起動できなかった理由として、仮想ルータのメモリが足り

ずにその処理能力以上のパケットが流れ続けているからではないかと考えられる。また、PCの動作が鈍くなり反応が遅くなった。

表1 B方式での主・副ノードの結果

	NS3	NS4	NS5	NS6	NS7
主ノード		3	3	3	3
主ノードに対する値		0.050	0.050	0.050	0.050
副ノード			4	4	4
副ノードに対する値			0.100	0.100	0.100

	NS8	NS9	NS10	NS11	NS12
主ノード	3	8	8	8	8
主ノードに対する値	0.015	0.100	0.100	0.100	0.100
副ノード	4	3	9	9	9
副ノードに対する値	0.200	0.150	0.100	0.100	0.100

表1は、それぞれのNS(ノード)がどのノードを主ノード、副ノードとするかを示し、各ノードに対する値を示したものである。その値とは、3.1の式(2)で計算された値のことであり、値の小さいノードから順に主ノード、副ノードとしている。

表2 ノード10個での性能比較

	B方式		A方式	
最大 P2P 中継回数	1		0	
平均 P2P 中継回数	0.44		0	
最大 iphop 数	3		3	
平均 iphop 数	1.22	1	4.11	2
最大 IX 通過回数	2		2	
平均 IX 通過回数	0.22	3	1.11	4

表3 ノード100個での性能比較

	B方式		A方式	
最大 P2P 中継回数	1		0	
平均 P2P 中継回数	0.49		0	
最大 iphop 数	3		3	
平均 iphop 数	1.02	5	2.01	6
最大 IX 通過回数	2		2	
平均 IX 通過回数	0.02	7	1.01	8

表2、表3は、B方式とA方式との性能の比較である。表2、表3を見ると、B方式の方が、最大、平均共にP2P中継回数の値は大きい。平均 iphop 数の値は、表2の1、2を見ると約3、表3の5、6を見ると約1小さい。また、平均 IX 通過回数の値は、表2の3、4を見ると約1、表3の7、8を見ると約1小さい結果になった。この結果より、考案されたルーティングでは、P2P中継回数の値が大きいため、ストリー

データの品質の劣化の恐れがある。しかし、平均 iphop 数および平均 IX 通過回数の値が小さい結果から、ネットワーク全体へのトラフィックは軽減される。よって、多少データは劣化するが、とにかく速くデータを受け取りたいときに、B方式でのP2Pが特に優れていることが分かった。

7 おわりに

本研究では、ノード間の接続の安定性を目的としたP2Pストリーミングにおけるルーティング方式の実装とその性能評価を行った。主ノード、副ノードのどちらかがノードを離脱したとき、接続するためのノードを新たに見つけ出し、接続し直すという機能の実装を目指した。しかし、主ノード、副ノードの離脱を各ノードに知らせ、ノードを新たに見つけ出すところまで実装できたが、新たなノードを主ノード、副ノードとしてつなぎ換えることは実現できなかった。

GINEを使ったエミュレーション実験により、B方式の方がIX通過回数が少ないために、ネットワークのトラフィックが軽減されるという結果を得た。また、仮想ネットワークモデルの違いによって違う結果が得られると予想される。

今後の課題としては、まずノードをつなぎ換えられるように実装を完成させる必要がある。また、副ノードを準備しているルーティング方法では、準備していないルーティング方法に比べ、ノードをつなぎ換えストリーミングを再開するまでにかかる時間が実際にどの程度改善されるのかを数値で確認する必要がある。今回我々は、仮想ネットワークを1層のみで作ってしまったので、複数の層を作ってエミュレーション実験を行うと、また違った結果が期待でき、より詳しい考察ができると考えられる。最後に我々は実際にデータを流して実験しておらず、データを実際に流しGINEによってパケットロスや遅延を設定して実験し、性能評価を行うことも課題の一つである。

参考文献

- [1] 後藤祐輝, 田中達也: P2Pストリーミングのためのルーティングの提案とそのエミュレーション, 卒業論文, 南山大学 数理情報学部 情報通信学科 (2007).
- [2] 片岡佑, 南川陽平, 中嶋拓実: P2Pストリーミングのためのルーティングの提案とそのシミュレーション, 卒業論文, 南山大学 数理情報学部 情報通信学科 (2005).
- [3] Sugiyama, Y. and Goto, K. (Eds. Zhang, S. et al.): Design and Implementation of a Network Emulator using Virtual Network Stack, Proc. of the Seventh International Symposium on Operations Research and Its Applications (ISORA2008), Lecture Notes in Operations Research, Vol.8, pp.351-358, World Publishing Corporation (2008).