

# 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

2004MT018 後藤 祐輝

2004MT106 田中 達也

指導教員 後藤 邦夫

## 1 はじめに

現在、P2P ストリーミングの利用が増え、P2P 技術の発達により PeerCast[1]、ShareCast[2] などが開発され、誰でも配信者となれる P2P ストリーミングが実現される。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、その下流のノードへのストリーミングが中断されてしまう。すると、再接続するための時間を要し、その間ストリームデータが受け取れない問題がある。また、既存のルーティング方法では IP 中継回数を重視し接続を行うので P2P 中継回数など多くなってしまふ。

本研究では、過去の卒論(発行 2005 年)[3] で考案された P2P ストリーミングのためのルーティングとそのシミュレーションの引継である。それを我々は、アプリケーションを作りネットワークを通して実験する。まず、ストリームデータを受信し受け渡す中継 P2P ノードのプログラムを作成した。そして、ノードにパラメータを設定しそれを考案した式で計算する。また、新規参加ノードが上流ノードを決める際に主ノード、副ノードの 2 つの上流ノードを決定する。実験では、GINE(Goto's IP Network Emulator)[4] を利用しノードに具体的なノード情報を与え新規参加ノードが接続するルーティングのネットワークの安定性について評価した。

後藤祐輝は主に実験環境構築、動作評価を担当、田中達也は主にプログラム作成を担当した。

## 2 既存の P2P とそのルーティングについて

Peer to Peer(以下、P2P)は、ネットワーク上の他のコンピュータ(ノードとも言う)に対してクライアントとしてもサーバとしても働くようなノードの集合によって形成されるものを指す。

### 2.1 ノード間の距離を考慮したルーティング方式

現在、一般的な PeerCast や、ShareCast の P2P ストリーミング方式では新規参加ノードから参加中ノードまでの hop 数でノード間のネットワーク距離を計算し、一番近いノードを選ぶ方式である。

上流ノードがストリーミングから離脱した場合、下流ノードは再び上流ノードを探索することになる。このためネットワーク全体にトラフィックが増加してしまう。また、参加中ノードが増えれば増えるほど再接続する時間がかかるようになる。

### 2.2 ノード間の接続を考慮したルーティング方式

PeerCast の P2P ストリーミング方式で、ノード間の接続に着目したルーティングとして各ノードが他のノード

との接続を維持するかどうかをストリーミング参加時に自己申告させて行うもの [5] がある。この方法でルーティングを行うと接続が途切れる箇所が従来よりも減らすことができるので全体的にみると安定した P2P 接続にできる。

- 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と申告したノード
- 一次参加ノード:接続を維持すると申告しなかったノード

このように 2 種類のノードに分ける。常時参加ノードには下流ノードに常時参加ノードを 1 つは接続するようにし、一時参加ノードには下流ノードを 1 つだけ接続する。しかし、この手法ではネットワークの距離は考慮されておらず、遠く離れたノード同士の接続が多くなる。

## 3 システムの概要

前述した既存の P2P ストリーミングにおけるルーティング方式の問題点を解決するための方式について説明する。

### 3.1 ルーティングモデル

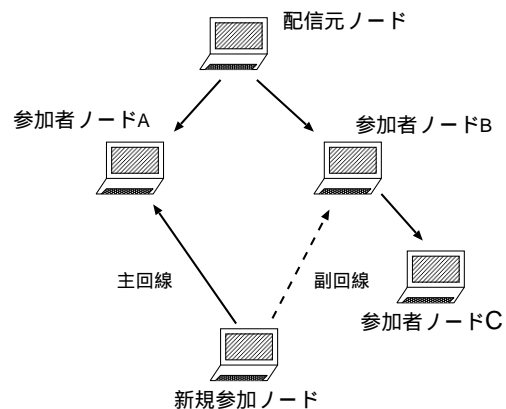


図1 ルーティングモデル

本研究では過去の卒論で考案されたルーティング方式を参考に行う。過去の卒論ではルーティング方式のシミュレーションまでだが、我々はパラメータを数式化して計算を行い、ノードを決定する方法を考案した。それに加えて、P2P ストリーミングにおける一番の欠点であるノードのストリーミング通信の離脱によって再接続するための時間を減らすために、接続先ノード以外にあらかじめ別の接続先ノードを決めておくという方法である。これにより接続が途絶えてしまってもすぐに接続し

直すことができるので検索時間を減らすことができる。

図1はルーティングモデル図である。主回線は参加者Aノードからストリーミング通信を行っている回線である。副回線は参加者Bノードから経路を確保するだけでストリーミング通信は行っていない。

### 3.2 システムの流れ

前提として全てのノードはどこか1つのノードのIPアドレス、UDPポートを知っている。こうすることでノードが起動時に番組表を受信することができる。番組表とは動画の内容とその動画の配信元ノードのIPアドレス、UDPポート、マルチキャストアドレスの情報の一覧である。そして、1つのマルチキャストアドレスに対して配信元ノードは1つとする。また、番組表では複数の番組がありP2P上でのみ意味をもつアドレスで表現する。そして、その中から一つの番組を選択し配信できる。P2Pノードは受信している番組以外は受信は行わない。経路表はそのノードに対して受信しているノード、送信しているノードのIPアドレス、UDPポートの情報である。副ノードに対して相手がまだ接続しているかどうか確認するため数秒間隔で通信を行い応答があるかを確認する。番組表と経路表はともに全てのノードに流れている。図2はシステムの流れ図である。

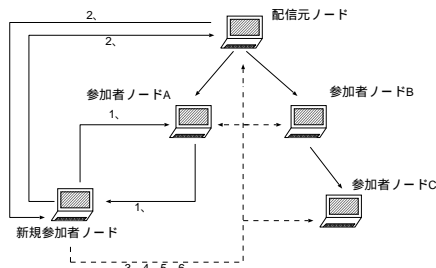


図2 システムの流れ図

1. 新規参加者ノードが起動するとまず、どこでも1つ既知のノードから番組表を受信する。番組の情報は...宛先マルチキャストアドレス、配信元ノード(IPアドレス、ポート番号)、動画の内容説明など
2. 番組表から番組を一つ選び、その動画の配信元ノードに接続し経路表を受信する。
3. 経路表を参照し配信元ノードから、配信先をたどりノード決定方法に従い主候補ノード、副候補ノード(P2Pノードが2個以上の場合)を決定する。
4. 主候補ノードに配送要求する。
5. 主候補ノードから送信を始める。経路表を更新する。
6. 受信ノードは主候補ノードからの配送が切れた場合、副候補ノードに配送要求し、これを主候補ノード、新たに副候補ノードを探す。

### 3.3 ノード決定方法

上流ノード決定に必要な各ノードのパラメータの情報を説明する。

- P2P中継回数(p2pcou)：配信元からのストリームデータのP2P中継回数( $p2pcou \geq 0$ )
- IP中継回数(ipcou)：新規参加ノードと接続先ノードとのiphop数( $ipcou \geq 0$ )
- 毎秒ごとに送るパケット(mainp)：ノードが下位ノードに毎秒ごとに送るパケット( $mainp \geq 0$ )
- 主ノードの数(mcou)：あるノードに対してストリームデータを受信している下位ノードの数( $mcou \geq 0$ )
- 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( $scou \geq 0$ )
- 初期設定のバンド幅(bands)kbps：各ノードが利用可能なバンド幅( $bands \geq 0$ )
- 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バンド幅( $bandw \geq 0$ )

これらのパラメータを与えた理由は、P2Pストリーミングでは各ノードでストリーミングデータをコピーし下流ノードへ渡すのでP2P中継回数が増えるとストリーミングデータの品質が劣化してしまうからである。IP中継回数は少ないほどデータ通信の時間が減らすことができる。また、各ノードにおいてストリーミングデータを送信可能な帯域が少ないとそこで遅延が発生するのでその値を求められるようなパラメータを用意した。

パラメータの情報をノード情報とし、新規参加ノードが参加中ノードに接続する際にノード情報を使用し主、副候補ノードを決定する。考案した式を以下に示す。

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

$$x = \alpha \times p2pcou + \beta \times ipcou + \gamma \times \frac{1}{bands - mainp - bandw \times scou} \quad (2)$$

$$(x > 0)$$

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

(1)式を満たすノードから主、副ノードを選ぶ。(1)式はあらかじめ各ノードで設定している利用可能帯域からストリーミング通信や制御のための通信に使用している帯域を引いた残りの帯域から、さらに副回線の送信帯域を引いた残りの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ことでストリーミング通信ができる。この(1)式を満たすノードは新たな下位ノードにストリームデータを送信することができる。

(1)式を満たす場合、そのノード情報を(2)式で計算する。(2)式は候補ノードの中から主、副ノードの2つのノードを決定するための式である。P2P中継回数、IP中継回数、残り利用可能帯域の3つの値を重みつき平均を行い足すことによりxの数値が一番小さいノードを主ノード、次に小さいノードを副ノードに決定する。 $\alpha, \beta, \gamma$ は重みつき平均である。必要バンド幅が最低限ないとストリーミング通信が行えないので必要バンド幅を

一番重視し、次に P2P 中継回数は数が増えるとデータの劣化や遅延が発生するので次に重点を置く。そして、IPhop 数は少ないほどネットワーク全体のトラフィックが軽減できるが小規模での P2P ストリーミングを考慮しているためさほど重視しなかった。これらを重点的に考慮した結果重み付きの数値は  $\alpha=0.12$ ,  $\beta=0.03$ ,  $\gamma=0.85$  に設定した。

#### 4 システムの実現

この節ではシステムの実現の詳細について述べる。提案するシステムは 3 つのプログラムによって構成される。

- streamer : UDP でストリームを発生するプログラム
- P2P ノード : streamer から送られてきたデータを制御し受け渡すプログラム
- listener : P2P ノードから送られてきたデータを再生するプログラム

streamer, listener は外部にあるプログラムであり、動画や音声を直接送信・再生するプログラムである。P2P ノードはメインプログラムであり、この P2P ノードがそれぞれつながることで P2P ストリーミングが行われる。

##### 4.1 P2P ノードについて

P2P ノードの概要について説明する。

- 中継スレッド : データの送受信を行う
- 制御スレッド : ノード決定方法をもとに経路を決定する
- 経路表 : 制御スレッドによって決定された経路情報を書き込んでおく

制御スレッドは他の P2P ノードの制御スレッドと通信を行い接続ノードを決定し、経路表に書き込む。そして、中継スレッドは経路表を参照して送られてきたストリーミングデータを次のノードに送る。このような流れで P2P ノードは動作する。また、このシステムはソケットを 3 つ使用し、ポートを 2 つ bind する。ソケットを 3 つ使う理由は制御スレッド用と中継スレッドの受信と送信用である。また、ポートが 2 つある理由も制御スレッドと中継スレッドで別々に通信用ポートを確保するためである。

#### 5 実験環境

この節では実験環境の詳細について述べる。

図 3 は実験モデル図である。

##### 5.1 ネットワークモデル

実際にすべての P2P ユーザを動作させて実験を行うのは困難なので送信ノードと受信ノードは実際にネットワーク上で動かす、P2P ノードは GINE を利用してエミュレートする。GINE ではルータの設定を行い最適なノードに接続するか実験をする。実ネットワークで

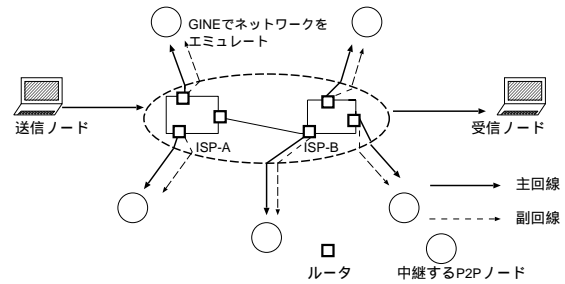


図 3 実験モデル図

は複数の ISP (インターネットサービスプロバイダ) があり、ISP 間で IX (Internet eXchange) を行い通信を行う。ISP 間での通信が増加すると IX に負担がかかり、ネットワーク全体に影響が及ぶ。提案ルーティング方式でこの問題が解決できるか調べるために 2ISP の小規模な利用モデルで PeerCast の既存方式にみられる IPhop 数が最小ノードを上流に選択する方法と比較する。

実験では、ISP-A のノード 10 個、ISP-B のノード 10 個から構成される IP ネットワークを構成し、提案した方式に基づき P2P ルーティングを行う。それと既存のルーティングと評価した。ここでの既存のルーティングとは IPhop 数だけを参考にして接続ノードを決定する方法である。同じ ISP 間のノードの IPhop 数は 1 から 10 個、2ISP 間のノードの IPhop 数は 10 から 15 個に設定した。そして、各ノードの利用可能帯域を設定する範囲は 100(0.1Mbps) から 1000(1Mbps) までとする。この条件で IPhop 数と利用可能帯域をランダムに設定し実験を行った。

##### 5.2 実験結果

実験は実験環境に基づきプログラムを動かして行うのが本望だが実際実験を行うことができたのは P2P ノードを接続しそこに音声データをストリーミング形式で流したときの動作確認と中継性能である。CD 音声ファイルを 1.4Mbps(1400 オクテット、126 packets/sec) で UDP 送信した結果スムーズなストリーミングができた。よって以下の結果はプログラムのルーティング部分を用いて行ったシミュレーション結果である。

それぞれのルーティングによって接続したノードの結果を図 4 と 図 5 で示す。尚、ノードの番号は 1 が配信元ノードであり、それ以降はノードの番号順に接続を行った。

図 4 より IX の通過は木構造の上位層で起こっていることが分かる。これは参加中ノードが少ないときは同じ ISP 内に最適なノードが存在せず、他方の ISP 内のノードに問い合わせるためと考える。また、ISP 間のデータの往来がないのでさほどネットワーク全体に負荷がかかることはないと考えられる。図 5 では IX の通過は一つしか通っていないことが分かる。既存のルーティングは IPhop 数だけを着目したノード決定方法なので同じ ISP 内のノードは他方の ISP 内のノードより IPhop 数

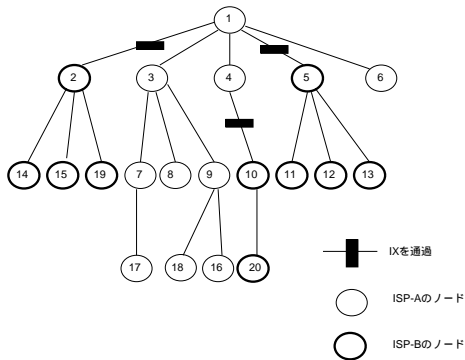


図4 考案方法によるルーティング結果

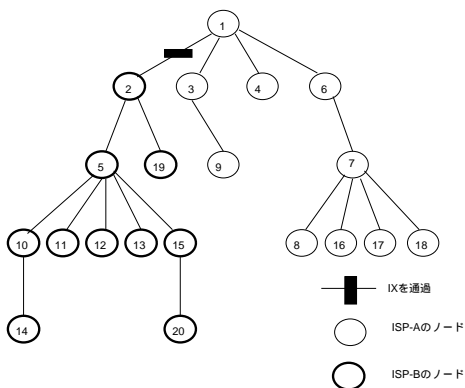


図5 既存方法によるルーティング結果

を小さく設定しているため、同じISP内のノード同士でつながるためと考える。

表1 性能比較

	考案方法	既存方法
最大 P2P 中継回数	2	3
平均 P2P 中継回数	0.95	1.47
最大合計 IPhop 数	12	15
平均 IPhop 数	8.2	9.95
利用可能帯域が 0 未満のノード数	0	2
IX 通過回数	3	1

表1は考案したルーティングと既存のルーティングの性能の比較である。平均P2P中継回数は表から考案した手法のほうが小さいことが分かる。このことよりP2Pの中継回数を減らし、ストリームデータの品質の劣化を防ぐことが可能だと考える。ストリームデータの品質の劣化を防ぐことが可能だと考える。また、平均IPhop数は表から考案した手法のほうが小さいことが分かる。このことより既存よりネットワーク全体へのトラフィックを軽減できる。残り利用可能帯域では数値が0未満のノードは既存では2つに対し考案では0である。これに

より既存よりノードへのトラフィックを軽減できる。

この結果より考案したルーティング方式は、動画像の品質の高い安定したストリーム配信が可能だと考える。

## 6 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提案を行った。ノードにパラメータを設定し他のノード情報と組み合わせることにより、最下層のノードまでP2P接続の安定性が向上するようになった。ルーティング実験の結果、次のような結果を得た。

- P2P 中継回数の縮小
- 上流ノード離脱時から再接続までの時間の短縮
- 一つのノードにかかる負荷の軽減

P2P中継回数が減ることにより、品質の劣化を抑えることができ品質の高い状態でのP2Pストリーミングが可能となった。主ノード、副ノードと2つの上流ノードを決定することですぐにストリーミングを再開することが可能となり安定したストリーミング通信ができた。また、残りの利用可能帯域幅を重視して接続先ノードを決定することで既存の方法では一つのノードにおいて過剰に接続してしまいネットワークの遅延とパケットロスの原因になってしまいが考案方法では過剰に接続することを減らすことができその原因を抑えることができた。

本研究での実験はプログラムを作成しその動作確認に留まった。今後の課題としてGINEによって遅延とパケットロスを発生させたときの実験と性能評価がある。また、ノード数をさらに増やしたときどのような結果になるか、主ノードの切替えによって発生するトラフィックの測定、P2Pネットワークにおけるスループットの測定が必要である。

## 参考文献

- [1] PeerCast: 初心者向け PeerCast 解説, <http://peerccast.gooside.com/>
- [2] ShareCast: ShareCast ストリーミング配信システム, <http://www2h.biglobe.ne.jp/hnakamura/technolab/howto/sharecast/sharecast.html>
- [3] 片岡 佑, 南川 陽平, 中島 拓実: P2P ストリーミングのためのルーティングの提案とシミュレーション, 南山大学数理情報学部情報通信学科, 卒業論文, (2005年)
- [4] Ihara, A., Murase, S. and Goto, K.: IPv4/v6 Network Emulator using Divert Socket, *Proc. of 18th International Conference on Systems Engineering(ICSE2006)*, Coventry, UK, pp. 159-166(Sep.2006)
- [5] 藤崎 貴章, 川島 幸之助: ピアの離脱を考慮したP2P型ストリーム配信のネットワーク構成, 信学技報, NS, Vol.103, No.10, pp. 5-8 (200304)