

異種ネットワーク環境でのP2Pストリーミング

小 俣 栄 治^{†1} 石 川 憲 洋^{†1} 村 上 慎 吾^{†2}
ヨハン イェルム^{†3} 泉 知 論^{†4} 中 村 行 宏^{†5}

近年、インターネットの急速な発展にともない、コンテンツの分散管理が可能な P2P 技術が注目されている。さらに、大規模な配信サーバが不要である、負荷分散が可能であるなどの利点から、P2P 技術を用いたマルチメディアコンテンツのストリーミングの需要も高まっている。また、現状では、マルチメディアコンテンツはインターネットに限らず、様々なネットワーク上に分散している。そこで本稿では、インターネットに限定せず、ホームネットワークやアドホックネットワークなどが混在する異種ネットワーク環境において P2P ストリーミングを実現する方式を提案する。さらに、本 P2P ストリーミングのプロトタイプの実装と評価について述べる。

Peer-to-Peer Streaming over Heterogeneous Networks

EIJI OMATA,^{†1} NORIHIRO ISHIKAWA,^{†1} SHINGO MURAKAMI,^{†2}
JOHAN HJELM,^{†3} TOMONORI IZUMI^{†4} and YUKIHIRO NAKAMURA^{†5}

Recently, P2P technology that manages a lot of contents in a highly distributed manner is paid to attention with a rapid development of the Internet. A demand for the P2P Streaming of the multimedia content has risen due to the advantages that a large-scale delivery server is unnecessary and a load-balancing is possible. Multimedia content are currently distributed over networks including the Internet and the home network. Therefore, we propose a method of P2P Streaming for heterogeneous network environment. We then discuss the prototype implementation and the performance evaluation of P2P Streaming.

1. はじめに

近年、インターネットの急速な発展にともない、多くのコンテンツがネットワーク上に存在するようになっており、従来の Server/Client モデルによるコンテンツの集中管理が困難になりつつある。そのことから、コンテンツを分散的に管理する Peer-to-Peer (P2P) 技術が注目を集めている。グループウェア¹⁾ や分散検索システム²⁾ などの新しい形の分散コンピューティングアプリケーションや、JXTA³⁾ のような汎用的な P2P プラットフォームなどの研究・開発がさかんに行われている。

また、ユーザ環境の発展にともない、ユーザ自身が大容量のマルチメディアコンテンツを取り扱うことが可能になりつつある。そのような背景から、P2P 技術を用いたマルチメディアコンテンツのストリーミング技術 (P2P ストリーミング) が注目を集めている。P2P ストリーミングは、各ユーザがリソース (CPU やネットワーク帯域など) をシェアして行うことから、大規模なサーバを必要とせず、さらにネットワークの負荷を軽減させることが可能である。また、マルチメディアコンテンツは、インターネットに限らず様々なネットワークに分散している。

そこで本研究では、インターネット、ホームネットワークなどが混在する異種ネットワーク環境において P2P ストリーミングを実現することを目的とする。

本稿の構成は、以下のとおりである。2 章において提案する P2P ストリーミングのアーキテクチャについて述べる。3 章ではこれまでに提案している P2P プラットフォームの概要を示し、4 章では本プラットフォーム上で設計したプロトコルとそのプロトタイプ実装について述べ、評価結果を示す。5 章では考察を述べ、6 章に本研究の関連研究をあげる。最後に 7 章

†1 株式会社 NTT ドコモ
NTT DoCoMo, Inc.

†2 株式会社日本エリクソン
Nippon Ericsson K.K.

†3 エリクソン・リサーチ
Ericsson AB

†4 立命館大学
Ritsumeikan University

†5 京都大学
Kyoto University

でまとめを述べる。

2. P2P ストリーミングアーキテクチャ

P2P ストリーミングとは、ストリーミングデータをパケットリレー方式で転送する仕組みである。P2P ネットワーク上のすべてのノードは、配信・中継・受信の役割を担う。

本研究では、P2P 技術を用いたストリーミング方式において、特に、インターネット、ホームネットワークなどが混在する異種ネットワーク環境においてストリーミングを実現するための提案を検討する。

2.1 要求条件

異種ネットワーク環境での P2P ストリーミングを検討するにあたり、以下の項目を要求条件とした。

様々なネットワークへの対応

異種ネットワーク環境で想定されているような IP (インターネットなど)、non-IP (IEEE1394 など) が混在する環境において、シームレスなストリーミングを実現することが求められる。さらに、各ネットワークに最適化されている既存のストリーミングプロトコルの効果的な利用も重要である。多様な通信形態への対応

様々な通信形態 (1 対 1, 1 対多) に対応したうえで、効率的にストリーミングデータを配信することが求められる。

耐障害性

P2P ノード間の故障、離脱が起こる環境であっても、中断することなく、ストリーミングを維持することが求められる。しかし、各資源は有限であるため、資源の利用は最小限に抑えることが重要である。

2.2 異種ネットワーク環境への対応

本研究では、様々なネットワークが混在する環境において、P2P ストリーミングを実現することを目的としている。インターネット、ホームネットワークなどが混在し、それらのネットワークをオーバーレイネットワークである P2P ネットワークにより接続する。このような環境において P2P ネットワーク上で、シームレスにストリーミングデータを転送する方式を検討する。

P2P ストリーミングでは、経路などを管理するための制御メッセージの転送とその経路に沿ってストリーミングデータを転送する処理が必要である。制御メッセージとストリーミングデータの転送方式として、大きく分類して以下の 2 つが考えられる。

方式 1: 制御メッセージとストリーミングデータを、

同一プロトコルで転送する方式

方式 2: 制御メッセージとストリーミングデータを、独立したプロトコルで転送する方式

方式 1 では、制御メッセージとストリーミングデータを、同一プロトコルで転送することにより、制御メッセージとストリーミングデータの対応付けを行う必要がない。しかし、アプリケーション層プロトコルである P2P プロトコル上でストリーミングデータも転送することになるため、オーバーヘッドが大きくなってしまふ。また、ネットワーク環境に依存せず、すべて統一の転送プロトコルを用いるため、各ネットワークに特化して最適化されている RTP (Real-time Transport Protocol) や IEEE1394 Isochronous 転送のような、既存のストリーミングプロトコルは利用できない。方式 2 では、制御メッセージとストリーミングデータを、独立したプロトコルで転送する。そのため、制御メッセージは P2P プロトコルで転送、ストリーミングデータはストリーミングプロトコルで転送というように、転送フローを分割することが可能である。すなわち、ストリーミングデータについては、既存の各ネットワークに特化して最適化されたプロトコルを用いることが可能となる。しかし、制御メッセージとストリーミングデータのフローを分離することにより、制御メッセージとストリーミングデータの関連付けを与えるための処理が必要となる。本研究では、オーバーヘッドを最小限に抑え、効率的なストリーミングを実現するという観点から、制御メッセージとストリーミングデータを分離して転送する方式 (方式 2) を採用する。方式 2 を実現するため、2.3 節において制御メッセージとストリーミングデータの関連付けを与えるための処理を、以下のステップに分けて検討する。ここで述べた制御メッセージは、各ステップに必要な処理に応じて、新たに定義する。

- (1) 経路の設定: ストリーミング開始時にストリーミングデータの転送経路を設定する。
- (2) 経路の維持: ストリーミングデータ転送中の転送テーブルの管理、経路障害時の回復処理などを行う。
- (3) 経路の開放: ストリーミング終了時の経路開放を行う。

2.3 多様な通信形態への対応

提案する P2P ストリーミングでは、多様な通信形態に対応するために、ユニキャスト方式とマルチキャスト方式の 2 方式について検討する。ユニキャスト方式とは、ストリーミングデータを送信するノードと受信するノードが 1 対 1 の場合のストリーミング方式

である。また、マルチキャスト方式とは、ストリーミングデータを送信するノードが1ノードであるのに対し、受信するノードが複数存在するストリーミング方式である。以下に、各々の経路設定・維持・開放処理について述べる。

2.3.1 ユニキャスト方式

2.3.1.1 経路の設定

ユニキャスト方式の場合、送信ノード主導型のストリーミングと、受信ノード主導型のストリーミングが考えられる。そこで、上記2つのストリーミング形態に対応したユニキャスト方式について検討を行う。

まずはじめに、放送型ユニキャスト方式について検討を行った。提案するP2Pストリーミングでは、各ノードがどのような経路でどのデータを転送するのかをあらかじめ定める必要がある。そのため、ストリーミングを開始する前に、制御メッセージを用いてルーティングテーブルを作成する。ルーティングテーブルの作成には、以下の点を考慮する必要がある。

- 各ノードは、ストリーミングフローを一意に識別できなければならない。
- ノード間のトランスポートプロトコルは、そのリンクごとに決定する必要がある。
- 各ノードは、転送先のトランスポート層プロトコルに応じた待ち受けアドレスを知っていなければならない。

まず、ストリーミングフローを一意に識別するために、ID(フローID)を定義する。このフローIDは、送信ノードがストリーミングを開始する際に、ストリーミングフローごとに生成し、中継ノード、受信ノードに対して通知する。その結果、転送経路上の各ノードは、ストリーミングフローを一意に識別することが可能となる。

さらに、ストリーミングデータの送信には、送信先ノードのアドレスが必要である。トランスポート層プロトコルがUDP/IPの場合、アドレスはIPアドレス+ポート番号となる。しかし、ポート番号の範囲は制限されているため、送信ノードがポート番号を一意に決定すると他のストリーミングフローのポート番号と重複してしまう可能性がある。そこで、制御メッセージを受信した経路上の各ノードは、ノード内で一意となるポート番号を自ら決定する。決定したアドレスと転送プロトコルを各ノード間で設定するために、制御メッセージは、各ホップ(送信ノード→中継ノード、中継ノード→中継ノード、中継ノード→受信ノード)ごとに転送される。

以上の検討に基づく経路設定フローの例を図1に示

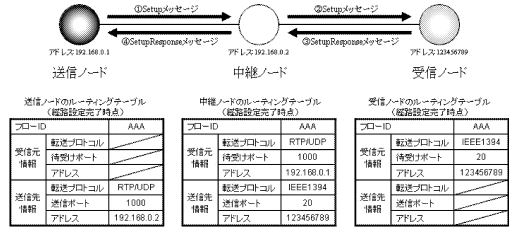


図1 ユニキャストの経路設定
Fig.1 Route setup of unicast.

```
<Setup xmlns="http://www.nttdocomo.co.jp/p2p_stream_ctl">
  <Source>975870cd-f9bb-4305-1234-e65e29f6029</Source>
  <Destination>
    <Route Node="975870cd-f9bb-4305-1234-e65e29f6030">
      <Target>975870cd-f9bb-4305-1234-e65e29f6031</Target>
    </Route>
  </Destination>
  <Flow ID="00001.2003-12-14T12:13:24@975870cd-f9bb-4305-1234-e65e29f6029">
    <Type>Broadcast</Type>
    <TransmissionProtocolID>rt</TransmissionProtocolID>
    <Bandwidth>384</Bandwidth>
    <TransmissionMedium>udp</TransmissionMedium>
  </Flow>
</Setup>
```

図2 Setupメッセージの例
Fig.2 Example of Setup message.

```
<SetupResponse xmlns="http://www.nttdocomo.co.jp/p2p_stream_ctl">
  <Result>Success</Result>
  <Flow ID="00001.2003-12-14T12:13:24@975870cd-f9bb-4305-1234-e65e29f6029">
    <Type>Broadcast</Type>
    <TransmissionProtocolID>rt</TransmissionProtocolID>
    <Bandwidth>384</Bandwidth>
    <TransmissionMedium>udp</TransmissionMedium>
  </Flow>
  <TransmissionAddress>192.168.0.2:1000</TransmissionAddress>
</SetupResponse>
```

図3 SetupResponseメッセージの例
Fig.3 Example of SetupResponse message.

す。中継ノードは複数存在してもよい。送信ノードは、各ノードに対して経路設定を要求するための制御メッセージを送信する。この制御メッセージを、Setupメッセージと定義する。さらに、Setupメッセージを受信したノードは、自ノードが希望する転送プロトコルなどのパラメータを持つ制御メッセージで応答する。この制御メッセージを、SetupResponseメッセージと定義する。定義した2つのメッセージの例を図2、図3に示す。一連の流れを行うことにより、各ノードは送信されてきたストリーミングデータをどのノードに対して転送するかを定めることができ、ストリーミングデータ転送が可能となる。また、オンデマンド型のユニキャスト方式は、最初に受信ノードが送信ノードに対してストリーミングの開始要求を行うことにより実現する。このストリーミング開始要求とその応答のための制御メッセージをそれぞれ、DeliveRequestメッセージ、DeliverResponseメッセージと定義する。続く経路設定処理は、放送型ユニキャスト方式と同様である。

2.3.1.2 経路の維持

P2P ネットワークはノードの参加・離脱が頻繁に発生するため、ストリーミング実行中に転送経路が切断されていないかを確認する必要がある。そこで、提案する P2P ストリーミングでは、経路の切断をすばやく検知するために、ソフトステート管理方式を用いる。各ノードは経路設定の状態をタイマにより管理し、タイマが切れる前に隣接するノードに対して状態維持を要求する。この要求を受けた各ノードは、タイマをリセットする。ソフトステート管理をすることにより、転送経路の切断や P2P ネットワークの輻輳により経路維持のための制御メッセージが届かなくても、各ノードは予約資源を解放できる。そのため、利用資源の消費を最小限に防ぐことが可能である。上記の経路維持を行うための制御メッセージを、接続状態確認メッセージである Keepalive メッセージと、その応答メッセージの KeepaliveResponse メッセージと定義する。経路の切断を検知した後の経路回復処理については、2.4 節で後述する。

2.3.1.3 経路の開放

ストリーミングの実行中、もしくは終了時に、ストリーミングの終了を要求するノードはフロー ID を指定して、設定経路を解放する。ストリーミングを終了する状況としては、以下の場合が考えられる。

- 送信ノードによるストリーミングの終了
- 送信ノード以外によるストリーミングの終了

上記の経路開放処理を行うための制御メッセージとして、送信ノードがストリーミング終了を通知する際に送信する Release メッセージ、送信ノード以外の受信ノード、中継ノードがストリーミング終了を通知する際に送信する Abort メッセージを定義する。

2.3.2 マルチキャスト方式

2.3.2.1 経路の設定

これまでに我々が提案している P2P プラットフォーム⁴⁾の P2P マルチキャスト⁵⁾は、各ノードが相互に接続し、直接的あるいは間接的に通信ができる P2P ネットワーク上で、ネットワークに参加する一部のノードをサブグループとして、効率的なグループ内通信を実現するものである。P2P プラットフォームの概要は、3 章で述べる。

本方式では、図 4 に示すように、P2P ネットワークにオーバーレイする形でマルチキャストツリーを構築する。マルチキャストツリーは、マルチキャストメンバノードに加え、マルチキャストメッセージのルーティングのみを行うルーティングノードによって構築する。また、新たに参加するノードは、そのツリーの

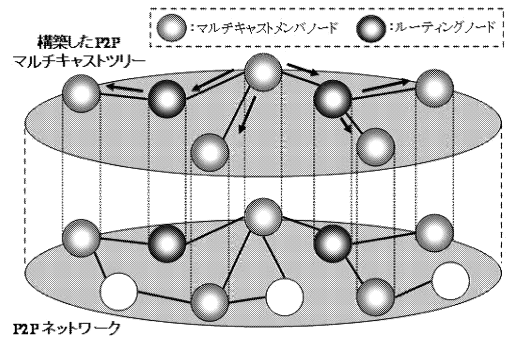


図 4 P2P マルチキャスト
Fig. 4 P2P multicast.

形状に対してマルチキャストメンバノードだけでなくルーティングノードを含めた最適な接続先ノードを選択し、接続することによってグループに参加する。このようにマルチキャストメンバノード以外のノードも含めてツリーを構築することにより、効率的なツリーを構築する。P2P マルチキャストツリー上を転送されるマルチキャストメッセージは、P2P ネットワーク上ではホップ・バイ・ホップにより転送される。P2P マルチキャストツリーの各メンバは、自ノードに転送されてきたマルチキャストメッセージを、メッセージを転送してきたノード以外の隣接ノードに転送する方式により、マルチキャストメッセージ配信を実現している。

P2P プラットフォーム上の P2P マルチキャストと、ユニキャスト方式との整合性をふまえたうえで、P2P マルチキャストストリーミングを検討する。マルチキャスト方式の経路設定には、以下の 3 つの方式が考えられる。

方式 1: ユニキャスト方式と同様の経路設定を行う方式

方式 2: P2P マルチキャストを利用して、経路設定を行う方式

方式 3: ユニキャスト方式と P2P マルチキャストを組み合わせた方式

方式 1 の場合、ユニキャストと同様の方式を用いる。ストリーミングデータの転送経路を設定するためには、どのようにストリーミングデータを転送するか（ストリーミングデータ伝送プロトコルや伝送媒体、帯域情報など）や前述したストリーミングフローを一意識別するためのフロー ID などのフロー情報を各メンバが知っている必要がある。しかし、ストリーミングを実施する前の段階では、送信ノードのみがその情報を知っているため、方式 1 の場合、送信ノードから経路の設定を行う必要がある。そのため、送信ノードの負担が大きくなり、マルチキャストの利点が損なわれ

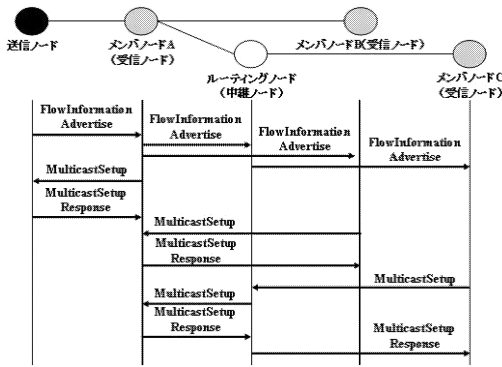


図5 マルチキャスト方式の経路設定
Fig. 5 Algorithm of multicast route setup.

てしまう．さらに、送信ノード主導で経路設定を行うことから、受信ノードが自由にストリーミングの受信と中断を行うことが困難になる．さらに方式2では、提案しているP2Pプラットフォームで規定しているP2Pマルチキャストを用いることになるが、経路設定を行う制御メッセージをP2Pマルチキャストで送信すると、マルチキャストメンバ以外のノード（ルーティングノード）は、経路情報を受け取ることができない．そのため、ストリーミングデータの転送テーブルを作成することができない．

そこで、上記の課題を解決するために、送信ノードがP2Pマルチキャストを用いて各マルチキャストメンバにストリーミングのフロー情報を制御メッセージを用いて送信し、その情報を受信した各マルチキャストメンバは、ユニキャストと同様な方法で送信ノードまでの経路を設定する方式3を採用する．ここで、フロー情報配信用制御メッセージ、経路設定の要求とその応答のための制御メッセージをそれぞれ、Flow-Information-Advertise メッセージ、Multicast-Setup メッセージ、Multicast-Setup-Response メッセージと定義する．図5に、提案する経路設定方式のアルゴリズムを示す．

- (1) 送信ノードは、ストリーミング配信に必要なフロー情報を、P2Pマルチキャストにより配布する．
- (2) 配信情報を受信したノードは、ストリーミングデータの配信を希望するかどうかを決定する．
- (3) ストリーミングデータの配信を希望するノードは、同一のフローIDで転送経路の設定が行われているかを確認する．
- (4) 経路が設定されていない場合は、受信したフロー情報と逆向きの方向に対して、経路設定を行う．

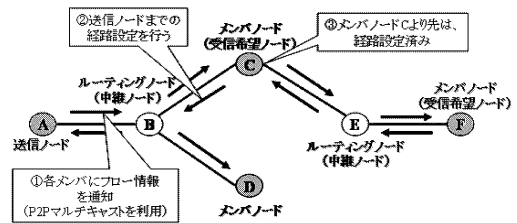


図6 マルチキャスト方式の経路設定の例
Fig. 6 Example of multicast route setup.

図6に、マルチキャスト方式経路設定の例を示す．ストリーミングを開始する送信ノードAは、グループメンバ全員にフロー情報を配布する．ストリーミングデータの受信を希望するノードCは、送信ノードAまでの経路設定処理を行う．経路設定要求を受信した送信ノードAは、ノードCの希望する設定情報をもとに、経路設定を完了させる．さらにノードFがストリーミングデータの受信を希望する場合は、すでにノードC-送信ノードA間の経路は設定済みのため、ノードCまでの経路を設定する．

2.3.2.2 経路の維持

2.3.2.1で述べたように、経路設定はユニキャスト方式と同様の方式を用いていることから、経路の維持についても、2.3.1.2で述べたユニキャスト方式と同様の経路維持を行う．ただし、マルチキャスト方式では、各ノードの接続状態が1対1だけではなく、1対多の場合もあるため、経路維持のための制御メッセージの交換も1対多に対応する必要がある．そこで、マルチキャスト方式の経路維持用制御メッセージとその応答メッセージとして、Multicast-Keepalive メッセージと Multicast-Keepalive-response メッセージを定義する．

2.3.2.3 経路の開放

P2Pマルチキャストストリーミングの終了には、以下の2つの方式が考えられる．

方式1：マルチキャストグループからの離脱を契機とする方式

方式2：P2Pストリーミングの終了を契機とする方式
方式1の場合、ストリーミングの終了は、すなわちマルチキャストグループからの離脱ということになる．しかし、マルチキャストツリーが必ずしもストリーミングのみで利用されているとは限らないことから、方式2のP2Pストリーミングの終了を契機とする方式が良いと考える．そこで、各ノードの役割ごとにP2Pストリーミングを終了する方式を検討した．

● 送信ノードからの終了

送信ノードは、隣接ノードに対してストリーミング終了を通知し、経路を開放する．さらにストリー

ミング終了を受信した各ノードも、ルーティングテーブルに従って、他のノードにストリーミング終了を通知する。この処理を繰り返すことにより、リーフノードまでのすべての経路を開放する。

- 受信ノード（中継ノード）からの終了経路を設定したノード（送信ノードと逆）方向に対して、ストリーミング終了を通知し、経路を開放する。ストリーミング終了を通知されたノードは、新たな経路を検索し、再設定を行う。
- 受信ノード（リーフノード）からの終了リーフノードである受信ノードは、送信ノード方向のノードに対してストリーミングの受信終了を通知し、経路解放処理を行う。

送信ノードからの終了、中継ノードである受信ノードから終了については、2.3.1.3 で定義した制御メッセージである Release メッセージ、Abort メッセージをマルチキャスト方式でも用いる。さらに、リーフノードである受信ノードからの終了する際の制御メッセージは、新たに Prune メッセージとして定義する。

2.4 耐障害性

P2P ストリーミングでは、中継ノードが離脱するために転送経路が切れてしまい、ストリーミングデータを転送できなくなる可能性がある。そのため、2.3 節で述べたように転送経路をソフトステートで管理し、経路の切断を検知した際には、他の経路に切り替える必要がある。そこで、ストリーミングデータの欠落を最小限に防ぎ、ストリーミングを継続するための方式を検討する。図 7 に示すような 2 つの基本的な方式を検討した。

図 7(A) の方式は、あらかじめ主経路と重複しない副経路を用意しておく方式である。中継ノード A が離脱したことにより経路が切れてしまった場合、即座に副経路に副経路に切り替える。この方式では、あらかじめ副経路を検索しておくことにより、すばやく切り替えることが可能である。しかし、副経路においても経路設定をしておくことになるため、資源の無駄遣いになってしまう。

図 7(B) に示す方式は、経路が切れてしまったノードから、再度経路を検索する方式である。中継ノード B が離脱したことにより経路が切れてしまった場合、上流の隣接ノードは受信ノードまでの経路を再検索する。この方式の場合、最小限の経路切替えを行うことにより、無駄な資源を省くことが可能である。しかし、受信ノードまでの経路の再検索と再設定が必要ことから、経路の切替えに時間がかかってしまう。

そこで本研究では、P2P ストリーミングをモバイル

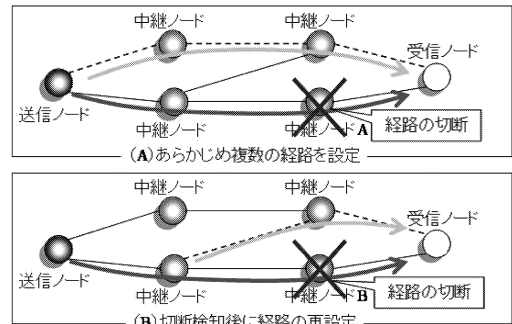


図 7 マルチパス
Fig. 7 Multi-path.

環境でも利用すること想定し、利用資源を最小限にとどめる方式 (B) を採用する。方式 (B) の場合、あらかじめ多くの資源を予約することを避けられるため、資源の無駄遣いを抑制することが可能である。

3. P2P プラットフォーム

3.1 アーキテクチャ

これまでに我々は、異種ネットワーク環境 (IP ネットワーク、IEEE1394 などの非 IP ネットワークが混在した環境) 上に、分散しているノード (携帯電話や PDA など) をシームレスに接続し、様々なアプリケーションを実行するための P2P プラットフォームの提案を行ってきた。

提案している P2P プラットフォームには、以下の構成要素がある。

P2P ノード: P2P ノードは独立した、双方向通信機能を持った構成要素を想定している。携帯電話や PC、情報家電など、あらゆるデバイスを想定している。

P2P ネットワーク: P2P ノードにより構成されたネットワークである。P2P ノード間では、相互信頼を前提として、接続を確立する。各 P2P ノードは、独立した構成要素であり、自由にネットワークに参加したり、離脱したりできる。

このように想定する P2P ネットワークは、異種ネットワーク環境に存在するデバイスが独立に P2P ネットワークに参加し、デバイス間でシームレスな P2P 通信を行うオーバーレイネットワークである。

3.2 プロトコル

これまでに提案している P2P プロトコルは、図 8 のような構造となっている。P2P Core Protocol は、マルチホップによるメッセージ転送の制御、ユニキャスト・マルチキャスト・ブロードキャストのサポートなどの基本的な機能を提供するプロトコルである。さ

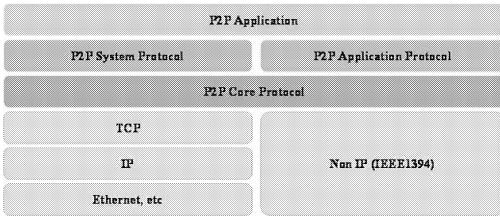


図 8 P2P プロトコルスタック
Fig. 8 P2P protocol stack.

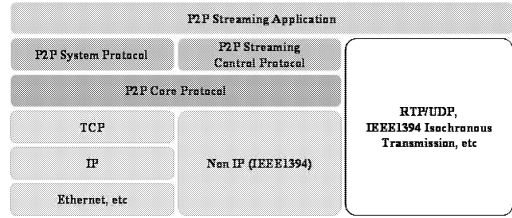


図 10 P2P ストリーミングプロトコルスタック
Fig. 10 P2P streaming protocol stack.

```

<Core xmlns="Namespace of P2P Core Protocol">
  <MsgType>Request</MsgType>
  <MsgID>12345.2002-12-20T16:15:32Z@968742ab-f9bb-4305-9900-f98e56f12352</MsgID>
  <Destination>
    <Target>874542ab-a5c6-4305-8745-f98e56f12347</Target>
  </Destination>
  <Source>968742ab-f9bb-4305-9900-f98e56f12352</Source>
  <ComType>Unicast</ComType>
  <MsgBody protocol="Namespace of P2P Communication Protocol">
    <Hello xmlns="Namespace of P2P Communication Protocol">
      <Mode>Hybrid</Mode>
      <RequestedCapability>
        <ExtendedProtocol ID=1>MulticastCommunication</ExtendedProtocol>
        <ExtendedProtocol ID=2>MulticastService</ExtendedProtocol>
      </RequestedCapability>
    </Hello>
  </MsgBody>
</Core>
    
```

図 9 P2P メッセージの例
Fig. 9 Example of P2P message.

らに、P2P メッセージをカプセル化して、TCP/IP や IEEE1394 を用いて送受信するためのフォーマットも定義している。また、P2P Core Protocol の上位プロトコルには、P2P System Protocol と P2P Application Protocol がある。P2P System Protocol は、アプリケーションに依存しない P2P ノード間の接続機能、マルチキャスト機能、セキュリティ機能などを提供するプロトコルである。P2P Application Protocol は、P2P Core Protocol や P2P System Protocol が提供する API を利用して、アプリケーションごとに必要な機能を定義するためのプロトコルである。すべてのプロトコルは、XML を用いて設計した。P2P メッセージの例を図 9 に示す。ここで示すメッセージは、P2P ノード間で P2P コネクションの接続要求を行うためのメッセージ（Hello メッセージ）である。

4. 実装と評価

4.1 P2P ストリーミングプロトコルの設計と実装
3章で紹介した P2P プラットフォーム上で、P2P ストリーミングプロトコルの設計を行った。2.2 節および 2.3 節の検討に基づき、図 10 に示すように、P2P ストリーミングのための制御メッセージを、P2P Streaming Control Protocol として定義した。P2P Streaming Control Protocol は、P2P Application Protocol の 1 つとして位置付けられる。また、ストリーミングデータの転送については、前述したように P2P プロト

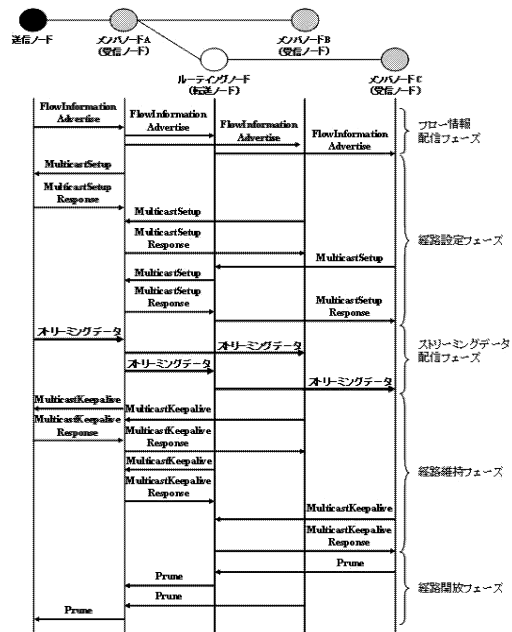


図 11 P2P ストリーミングメッセージシーケンス
Fig. 11 P2P streaming message sequence.

コルは利用せずに、既存のストリーミングプロトコル（RTP、IEEE1394 Isochronous 転送など）を用いて転送する。さらに、P2P Streaming Application を試作した。本アプリケーションは、P2P Streaming Control Protocol により転送され、ストリーミングデータ転送用の制御情報をルーティングテーブルに設定する機能、ルーティングテーブルに従ってストリーミングデータを転送する機能を提供する。RTP や IEEE1394 Isochronous 転送を利用したストリーミングデータの転送も、このアプリケーションにより管理する。

定義した P2P Streaming Control Protocol を用いて P2P マルチキャストストリーミングを実施する場合の経路設定から経路開放までのメッセージシーケンスを図 11 に示す。

- フロー情報配信フェーズ
送信ノードは、マルチキャストメンバノードに対

して、フロー情報を FlowInformationAdvertise メッセージにより通知する。

● 経路設定フェーズ

受信を希望するメンバノードは、FlowInformationAdvertise メッセージの逆向きの経路で MulticastSetup メッセージを送信する。MulticastSetup メッセージを受信したノードは、ストリーミングの可否を設定した SetupResponse メッセージを返信する。受信希望メンバノードから順にメッセージを交換し、各ノード間で必要な資源を予約しながら送信ノードまでの経路を確立する。メンバノード B の経路設定は、すでに送信ノードとメンバノード A 間の経路設定が終了していることから、メンバノード A までとなる。要求しているデータと同じデータ（フロー ID が同じ）の転送経路がすでに設定されている場合は、経路設定は行わない。

● ストリーミングデータ配信フェーズ

経路設定時に取り決めたプロトコルにより、各メンバノードに対してストリーミングデータを配信する。

● 経路維持フェーズ

転送経路維持のための MulticastKeepalive と MulticastKeepaliveResponse メッセージを各ノード間で定期的に交換する。

● 経路開放フェーズ

配信終了を要求するメンバノード C は、受信終了とともに隣接するルーティングノードに対して Abort メッセージを送信する。メッセージを受信したルーティングノードは、設定されていた経路を開放する。さらに、ストリーミングデータを受信していないルーティングノードがツリーの末端ノードとなることから、引き続きルーティングノードも隣接ノードに対して終了要求である Prune メッセージを送信する。

4.2 性能評価実験

本稿で提案した P2P ストリーミングの評価システムを試作し、性能評価実験を行った。評価実験は、IEEE802.3ab (1000base-T) でハブに接続した 5 台のパソコンを用いて行った。ネットワーク構成、各ノードの仕様を図 12 に示す。

まず、ユニキャスト方式の性能測定として、送信ノード A-中継ノード B-受信ノード C 間での経路設定時間と、中継ノード B が離脱した際の送信ノード A-受信ノード C 間での経路再構築時間を測定した。測定結果を表 1 に示す。さらにマルチキャスト方式の性能

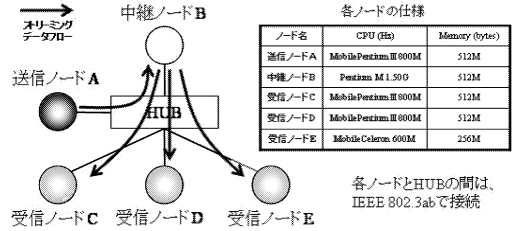


図 12 経路設定と再構築評価実験のネットワーク構成
Fig. 12 Network structure for route setup and re-construction.

表 1 図 12 における送信ノード A-受信ノード C 間の経路設定時間と中継ノード B が離脱したときの経路再構築時間

Table 1 Route setup and re-construction time between node A and node C in Fig. 12.

ノード名	経路設定の平均時間 (msec)	再構築の平均時間 (msec)
受信ノード C	1,266	3,291

表 2 図 12 における送信ノード A-受信ノード C, D, E 間の経路設定時間と中継ノード B が離脱したときの経路再構築時間

Table 2 Route setup and re-construction time between node A and node C or D or E in Fig. 12.

ノード名	経路設定の平均時間 (msec)	再構築の平均時間 (msec)
受信ノード C	1,269	6,179
受信ノード D	1,418	7,507
受信ノード E	1,565	9,374

測定として、送信ノード A-中継ノード B-受信ノード C, D, E 間における各々の経路設定時間と、中継ノード B が離脱した際の送信ノード A-受信ノード C, D, E 間における各々の経路再構築時間を測定した。測定結果を表 2 に示す。測定結果は、各処理を 10 回計測した際の平均値である。

マルチキャスト方式の経路設定はノード C, D, E の順に処理したことから、受信ノード C の経路設定時間は、ユニキャスト方式、マルチキャスト方式でほぼ同じである。しかし、中継ノード B の離脱時の再構築時間は、倍の時間がかかっていることが分かる。これは、分断を検知した受信ノード C, D, E がいっせいに再構築処理を行ったため、送信ノード A での処理が遅くなってしまったことによる遅延だと考えられる。また本実験では、受信ノード B はストリーミング終了を通知せずに離脱しているため、各受信ノードは 2.3.1.2 ならびに 2.3.2.2 で述べた維持メッセージにより分断を検知している。そのため、維持メッセージの交換のタイミングに応じて、再構築時間も変わってしまう。

続いて、TCP/IP ネットワークのみでの経路設定

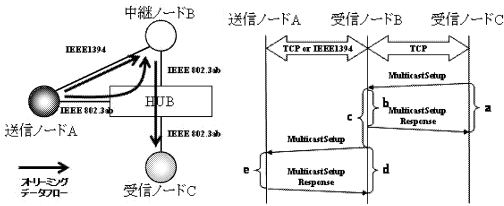


図 13 異種ネットワークをまたがった経路設定実験のネットワーク構成

Fig. 13 Network structure for route setup between different networks.

表 3 図 13 における送信ノード A-受信ノード C 間の経路設定時間

Table 3 Route setup time between sender node A and receiver node C in Fig. 13.

測定項目	TCP/IP 測定結果 (msec)	TCP/IP と IEEE1394 測定結果 (msec)
a	318	265
b	202	196
c	167	437
d	176	578
e	92	94

を行った場合と、TCP/IP と IEEE1394 が混在したネットワークで経路設定を行った場合の比較実験を行った。図 13 に測定のためのネットワーク構成と、測定項目について示す。送信ノード A と中継ノード B は、TCP/IP と IEEE1394 の双方で接続されており、この接続を切り替えることにより、比較実験を行う。測定項目は、a ~ e に示す処理に要する時間である。測定結果を表 3 に示す。測定した時間は、各処理を 10 回行った平均時間である。測定結果から、TCP/IP ネットワークと IEEE1394 が混在するネットワークでは、c と d の時間が大きく異なっていることが分かる。c は MulticastSetup メッセージの受信後、送信ノードに MulticastSetup メッセージを送信するまでの時間であり、d は MulticastSetup メッセージを送信してから MulticastSetupResponse を受信するまでの時間である。メッセージを処理している e の時間は、双方ともほぼ同じであることから、TCP/IP に比べ IEEE1394 はメッセージの送受信の速度が遅いことが分かる。遅くなる理由は、P2P メッセージをパケット化する処理が、TCP/IP に比べ IEEE1394 のほうが時間を要しているためと考えられる。

さらに、マルチホップとシングルホップストリーミングの場合で、パフォーマンス比較実験を行った。図 13 (TCP/IP ネットワークのみ) の送信ノード A-中継ノード B-受信ノード C 間と送信ノード A-受信ノード C 間におけるストリーミングデータのパケットロス

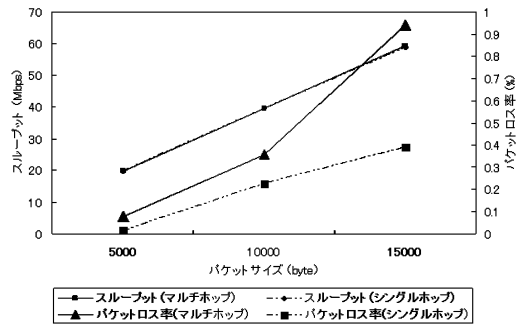


図 14 マルチホップとシングルホップのパフォーマンス比較

Fig. 14 Performance comparison of multi-hop and single-hop.

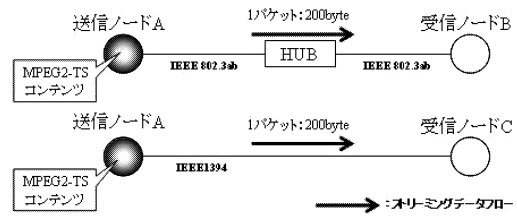


図 15 UDP/IP と IEEE1394 のパフォーマンス比較

Fig. 15 Performance comparison of UDP/IP and IEEE1394.

率とスループットを測定した。測定結果を図 14 に示す。本測定では、100 Mbyte のデータを各 5,000 byte, 10,000 byte, 15,000 byte に分割して送信した。測定値は、10 回の測定の平均値である。本実験環境で測定した場合、マルチホップ、シングルホップともに、スループットは同等の性能が得られることが分かった。また、マルチホップ、シングルホップともにパケットサイズを大きくすると、パケットロス率も大きくなる。マルチホップに関しては、10,000 byte を超えると急激に増加する。パケットロスはデータ受信時に起きていると想定されることから、マルチホップの場合、中継ノードでいったん受信していることによるパケットロスの増加であると考えられる。

最後に、UDP/IP と IEEE1394 で、同様のデータを送信した場合のパフォーマンス比較実験を行った。図 15 に示すような構成で実験を行った。配信するデータ形式はビデオ録画などで用いられている MPEG2-TS であり、データサイズは 180 Mbyte である。また、MPEG2-TS over IEEE1394 の規格上、MPEG2-TS の転送パケットサイズは 200 byte となっていることから、UDP/IP の場合も転送時のパケットサイズを 200 byte とした。実験の結果、10 回測定した平均スループットは、UDP/IP が 798.68 kbps であり、

IEEE1394 が 1,839.2 kbps であるという結果が得られた。IEEE1394 に比べ UDP/IP のスループットが減少してしまう理由は、パケットサイズが 200 byte という IEEE1394 に最適化されたサイズであることから、UDP/IP では転送処理が追いつかなくなってしまうという点に起因している。

5. 考 察

本稿で提案した P2P ストリーミングは、様々なネットワークへの対応、多様な通信形態への対応、耐障害性を要求条件として検討した。まず様々なネットワークへの対応については、制御メッセージとストリーミングデータのフローを分離することにより、既存のストリーミングプロトコルを活用したシームレスな P2P ストリーミング方式を提案した。さらに、表 3 に示す測定結果により、TCP/IP ネットワークと IEEE1394 ネットワークが混在した環境においてストリーミングが可能であることを確認した。しかし、TCP/IP に比べ IEEE1394 ネットワークでは、利用には影響のない範囲でありながらも、制御メッセージの転送に倍以上の時間を要している。原因としては、P2P メッセージをパケット化する処理時間の差に起因していると考えられる。実装した P2P ストリーミングシステムでは、IP over IEEE1394 などは利用せず、P2P メッセージを直接 IEEE1394 パケット化している。理由としては、以下の点があげられる。IP を利用することによりベストエフォート型の通信となってしまう、IP パケット処理によるオーバーヘッドのため IEEE1394 の性能を最大限に利用することができない。IEEE1394 が持つ QoS 機能が利用できなくなってしまう。さらに IP over IEEE1394 は、D-VHS などのデジタル家電への実装事例が少ない。このような点をふまえ、今後は IEEE1394 の機能を生かした制御メッセージ転送、ストリーミング配信について評価する。次に多様な通信形態への対応については、表 1 と表 2 に示す評価結果より、ユニキャスト方式とマルチキャスト方式のストリーミングが可能であることを確認した。本稿のマルチキャスト方式の実験では、5 ノードでマルチキャストツリーを構築した。今後は、さらに多くのノードでマルチキャストツリーを構築し、提案する P2P ストリーミングのスケラビリティについて評価する。耐障害性については、要求条件に基づき、資源の利用を最小限にとどめる方式を提案した。また、表 1 と表 2 の評価結果より、経路が分断された場合であっても経路を再構築してストリーミングを再開することが可能であることを確認した。しかし、中継ノードの離

脱による受信ノードの再設定時間は、各ノードにより異なってしまう。これはソフトステート管理による遅延だと考えられる。経路維持メッセージの交換時間を短くすれば、より早く経路分断を検知することが可能となるが、無駄なメッセージも多くなってしまう。経路維持メッセージを最小限に抑えながらも、より早く分断を検知する方式について検討する必要がある。また、図 14 に示す評価結果より、マルチホップ、シングルホップともに同等のパフォーマンスであることが分かった。しかし、中継ノードが増すほど、パケットロス率が増してしまう。また、図 15 の実験からも分かる通り、ネットワークが異なると最適なパケットサイズなども異なる。今後は、ネットワークの差異を吸収するためのバッファ、トランスコーディングなども含め、異種ネットワーク混在環境に適用可能な QoS 方式についての検討を行う。

6. 関連研究

P2P ストリーミングは、近年アプリケーション層マルチキャストとして研究されている。それらの中でも、ストリーミングを中心に検討されているものを示す。

PeerCast⁷⁾ は、Gnutella と同じ方式を用いてリアルタイムストリーミングを実現しているため、ネットワークを管理するノードは存在せず、各ノードが自立的に接続・切断を行う。ShareCast は、ベンチャー企業により開発されたミドルウェアである。ShareCast⁸⁾ では、認証や参加メンバ管理のためのサーバを用意している。参加メンバは、これらのサーバから情報を取得し、随時ネットワークを構築する。しかし、これらの研究は、共通してインターネットのみでの利用が前提となっている。関連研究との定性的な評価結果を、表 4 に示す。本研究は、我々が知る限り、インターネットだけでなく IEEE1394 などの非 IP ネットワークが混在する環境での P2P ストリーミングの実現を目指した最初の研究である。

表 4 関連研究との比較
Table 4 Comparison with relation research.

	提案方式	PeerCast	ShareCast
管理ノードの有無	無	無	有
配信ノード	非限定	非限定	限定
対象とするネットワーク	異種ネットワーク	インターネット	インターネット
セキュリティ機能	無	無	有

7. おわりに

本稿では、異種ネットワーク環境において P2P 技術を用いたストリーミング方式を提案した。本方式では、インターネットに限らず IEEE1394 ネットワークなどの様々なネットワークが混在する環境でストリーミングを行うことを目的としている。そのため、各ネットワーク媒体の特性を活かしたデータ転送を実現する方式をとって、ストリーミング制御チャネルとデータ転送チャネルを独立したプロトコルで実施する方式を採用した。本方式により、QoSなどを考慮した既存のマルチメディア転送プロトコルが利用可能である。また、様々なストリーミング形態の利用を考慮して、ユニキャスト方式ならびにマルチキャスト方式についての検討を行った。

今後は、携帯電話のような処理能力が低く、さらに利用可能なネットワーク帯域も少ない端末がノードとして参加した状況を想定し、QoSを考慮した経路検索方式の改良や、ストリーミングデータのトランスコードなどによる帯域の異なるネットワーク内の P2P ストリーミング方式などの検討を行う予定である。

参考文献

- 1) GROOVE. <http://www.groove.com/>
- 2) Gnutella. <http://gnutella.wego.com/>
- 3) JXTA. <http://www.jxta.org/>
- 4) Ishikawa, N., et al.: A Platform and Applications for Mobile Peer-to-Peer Communications, *www2003 workshop on Emerging Applications for Wireless and Mobile Access* (2003).
- 5) 加藤剛志ほか：ユニバーサル P2P プラットフォームにおけるシングルホップ P2P マルチキャストの検討，情報処理学会マルチメディア，分散，協調とモバイルシンポジウム (DICOMO 2004)，pp.277-280 (2004).
- 6) 木全哲也ほか：モバイル P2P ネットワークにおける AV 機器制御システムの構築，情報処理学会マルチメディア，分散，協調とモバイルシンポジウム (DICOMO 2004)，pp.449-452 (2004).
- 7) Zhang, J., et al.: Reliable End System Multicasting with a Heterogeneous Overlay Network, Technical Report GIT-CERCS-04-19, CERCS, Georgia Institute of Technology (2004).
- 8) ShareCast. <http://www.scast.tv/>

(平成 17 年 5 月 17 日受付)

(平成 17 年 11 月 1 日採録)



小俣 栄治

2002 年武蔵工業大学工学部電子情報工学科卒業。同年 NTT ドコモ入社。現在、NTT ドコモネットワークマネジメント開発部に所属。モバイルインターネットプロトコルおよびアプリケーションの研究開発に従事。



石川 憲洋 (正会員)

1978 年京都大学工学部情報工学科卒業。1980 年同大学大学院工学研究科情報工学専攻修士課程修了。同年日本電信電話公社 (現 NTT) 入社。現在、NTT ドコモネットワークマネジメント開発部に所属。モバイルインターネットの研究開発と国際標準化に従事。博士 (情報学)。電子情報通信学会会員。



村上 慎吾 (正会員)

1998 年筑波大学第 3 学群情報学類卒業。2000 年同大学大学院工学研究科修士取得後退学。同年日本エリクソン株式会社入社。以来、エリクソン・リサーチ・ジャパンにおいて、モバイルインターネットやアプリケーションに関する研究に従事。



ヨハン イェルム (正会員)

エリクソン・リサーチにおけるシニア・スペシャリスト。スウェーデン陸軍士官退役後、ジャーナリストとして 12 年間従事する。その間、雑誌の編集長としてインターネットに深く関わり、EU の On the Move プロジェクト等のコーディネータを務める。また、スウェーデンにおける最初のインターネットに関する本を出版し、スウェーデンのインターネットに初期の頃から深く関わる。エリクソンに入社後は、マサチューセッツ工科大学の客員研究員として在籍し、CC/PP 部会等の議長を務めて W3C フェローとなる。現在は Open Mobile Alliance のアーキテクチャ・グループの副議長も務める。



泉 知論 (正会員)

1969年生。1992年東京工業大学工学部情報工学科卒業。1994年同大学大学院理工学研究科電気・電子工学専攻修士課程，1998年同博士課程修了。博士（工学）。1998～2005

年まで，京都大学大学院情報学研究科通信情報システム専攻助手。2005年より立命館大学工学部電子情報デザイン学科助教授。1998年より株式会社シンセシス主幹研究員（兼業）。システム LSI の設計ならびに設計支援技術，再構成可能ハードウェアに関する研究開発に従事。電子情報通信学会，IEEE 各会員。



中村 行宏 (正会員)

1944年生。1967年京都大学工学部数理工学科卒業。1969年同大学大学院工学研究科修士課程修了（数理工学専攻）。同年日本電信電話公社（現 NTT）に入社。同社電気通信研

究所において，DIPS 論理装置の研究開発を経て，主に並列処理アーキテクチャを有するプロセッサの方式構成・設計技術の研究に従事。NTT 情報通信研究所高速通信処理研究部長等を経て，1996年9月より京都大学大学院工学研究科教授（電子通信工学専攻），1998年より同情報学研究科教授（通信情報システム専攻）。2004年より（財）京都高度技術研究所副所長（兼務）。工学博士（京都大学）。電子情報通信学会，IEEE 等会員。IEEE 関西支部役員，NPO パルテノン研究会運営委員長等。