

Universal P2P Architecture for feasible IP Multicast の設計と実装

中村 遼¹ , 岡田 耕司¹ , 重近 範行¹ , 村井 純¹

概要

本論文では、ユーザによる IP マルチキャストを用いた世界規模での自由な 1 対多型の通信を実現するため、Universal P2P Architecture for feasible IP Multicast (upamul) の提案を行う。IP マルチキャストはネットワーク上のホストグループに対して同報通信を行う仕組みであり、宛先が不特定多数であるストリーミングなどに適したアーキテクチャである。しかし現在のインターネットでは、様々な問題により IP マルチキャストを使用できないネットワークが多く、IP マルチキャスト可能なネットワークが離れ小島のように存在している。そのため、ユーザは自由に IP マルチキャストを送受信することはできない。そこで、upamul はエンドユーザを収容するエッジネットワークに既存の IP マルチキャストを用い、エッジネットワーク間の配送網をオーバーレイネットワークで構築することによって、ユーザの誰もが送信者となることができ、かつ受信者全てへのデータ配送を実現する。本論文では upamul のプロトタイプ的设计と実装を行い、実効帯域についての評価を行った。その結果、1000BASE-T で構築した実験環境において、限界実効帯域が 175Mbps であり、合計 130Mbps までであればストリーミングを実効的に行える事がわかった。

1 はじめに

インターネットの普及と広帯域化に伴い、動画や音楽などのリッチコンテンツのストリーミングサービスが増加している。現在その中でも特に、Ustream などをはじめとする、ユーザが 1 対多型での放送型リアルタイムストリーミングを行うサービスへの需要が高まっている。既存の多くのシステムでは、1 対多型の通信は、ユーザからのストリームを、専用に潤沢な資源の用意されたライブストリーミングサーバがユニキャストで各受信者へ中継することで実現している。しかし、既存のシステムがこのようなサーバ・クライアントモデルであるため、ユーザは特定の事業者によって用意されたサービスしか提供することができない。また、ユーザが 1 対多型の通信を実現できるような潤沢なネットワーク資源と計算機資源を用意することは難しいため、ユーザ自身で中継用サーバを用意することは現実的ではない。このように、既存のサーバ・クライアント型のシステムでは、ユーザがユーザの求めるサービスを自由に提供することはできないという問題がある。また、ユニキャストを前提としているため、受信者の増加にしたがって送信者の負荷が高くなるため規模性が無い。

上記の手法に対して、規模性のある 1 対多型の通信を実現するための手法として、IP マルチキャストやアプリケーションレイヤマルチキャストがある。IP マルチキャ

ストやアプリケーションレイヤマルチキャストでは、配送経路上のルータがパケットをコピーするため、送信者は単一のストリームを送信するだけで良い。そのため、受信者の増加にしたがって送信者側の負荷が上がることはない。しかし、双方ともに問題があり、ユーザによる自由な 1 対多型通信の実現には至っていない。IP マルチキャストは IP 層で 1 対多型の通信を実現するアーキテクチャである。IP マルチキャストは既に大抵のホストでサポートされており、ユーザはホストに変更を加えることなく、任意のアプリケーションを用いて IP マルチキャストを利用することができる。しかし、配送網の構築手法に起因するいくつかの問題があり、世界規模で実効的に動作してはいない。一方アプリケーションレイヤマルチキャストは、配送網をオーバーレイで構築することで 1 対多型の通信を実現する。配送網をオーバーレイで構築するため、バックボーンネットワークの構成に影響されることなく網を世界規模で拡張することができるという利点がある。しかし、配送網は構築するアプリケーションに依存したものになるため、他のアプリケーションがその配送網を汎用的に利用することはできない。

本研究では、既存手法の利点のみを活かして、世界規模で動作し、アプリケーション透過性と規模性を持つ 1 対多型の通信を実現するアーキテクチャとして upamul (Universal P2P Architecture for feasible IP Multicast) を提案する。upamul はユーザを収容するエッジネットワークに IP マ

¹慶応義塾大学

マルチキャストを用い、配送網をオーバーレイで構築する。現行のほとんどのホストはIPマルチキャストを使用可能であるため、それらのホストは特に変更を加えることなくupamulの構築するマルチキャスト網に参加することができる。upamulが利用するIPマルチキャストは、IP層の機能であるため、アプリケーションは透過的にこのマルチキャスト網を利用することができる。さらに、送信者は単一のストリームを送信するだけでよく、受信者数の増加に伴う負荷の増加も無い。また、配送網をオーバーレイで構築することによってバックボーンの構成に影響されることがなく配送網を広域に構築することができる。

本論文の構成は以下の通りである。次章で既存手法の問題点についてまとめる。3章では本研究における目的について述べる。4章では本研究に関連する研究についてまとめる。5章では提案手法の設計について詳述する。6章では帯域に関する実験と評価について述べる。最後の章では、評価から得られた結論と今後の課題についてまとめる。

2 問題点

既存のシステムであるIPマルチキャストとアプリケーションレイヤマルチキャストには多くの問題点があり、ともにユーザによる世界規模での1対多型通信を実現するには至っていない。IPマルチキャストの問題点のうち、技術的な問題として、現在IPマルチキャストのルーティングプロトコルとして広く利用されているPIMの問題点を以下に示す。PIMはモードによって構築する配送網が異なる。Dense Mode[1]では1つの送信者を基点として、複数の受信者へ向かう木構造の配送網を構築する。そのため送信者が複数の場合、送信者の数だけ配送網を構築する必要がある。よって網全体のルータの計算負荷が高くなるという問題点がある。もうひとつのモードであるSparse Mode[2]では、あらかじめ設定されたランデブーポイント(RP)を基点とした配送網を構築する。送信者はまずRPにマルチキャストパケットを転送し、RPはそのパケットのグループに応じて配送網へ転送していく。その後、RPを経由しない、最短経路の木構造へとトポロジーを移行していく。このように、PIM-SMでは全ての配送網においてRPが関与する。そのため、グループの数が増加するとRPの負荷が高くなるという問題点がある。

また、現在のIPマルチキャストには運用に起因する問題もある。IPマルチキャストを用いる既存サービスは、特定のコンテンツホルダーが特定のネットワークドメイン内

の受信者に対してのみコンテンツを配信するというモデルをとっている。ISP内のIPマルチキャスト網はこのサービスを実現するために設計されている。受信者を特定ネットワークドメイン内に限定するため、ドメイン内でのみIPマルチキャストを利用可能にし、ドメイン間でのルーティングを行わない。また、コンテンツを配信するサーバを基点とした配送網の構築しか行わないため、柔軟に送信者の追加を行えない。既存のIPマルチキャスト網がこのように、特定の送信ノードのみがマルチキャストデータを送信し、受信者を特定ドメインに限定する、という設計であるため、ユーザが自由にIPマルチキャストを用いてコンテンツを広域に配信することも、ユーザが自由に世界中のIPマルチキャストトラフィックを受信することもできない。

一方、アプリケーションレイヤマルチキャスト(ALM)は配送網をオーバーレイで構築するため、バックボーンネットワークの構成に影響されることがなく配送網を広域に展開することができる。しかし、ALMにも問題がある。ALMではユーザがインストールしたアプリケーションによって、1対多型の通信を行うための配送網が構築される。この配送網は、そのアプリケーションが行うサービスを実現するために最適化された配送網になる。そのため、配送網はアプリケーションごとに構築され、任意のアプリケーションが他のALMによって構築された配送網を利用することはできない。また、ユーザが新しいALMの配送網に参加するためにはその都度新しいアプリケーションをインストールする必要がある。このように、ALMにはユーザが求める複数のサービスを単一の網で構築するのが困難である、という問題点がある。ALMの問題点は4章に詳述する。

3 目的

本研究では、2章で述べたIPマルチキャストとALMの問題を解決し、ユーザによる世界規模での自由な1対多型の通信を実現することを目的としたシステムの構築を行う。この目的を達成するために必要なことは、少ない計算量で1対多型の通信を実現する配送網を構築すること、ユーザが自由にその配送網に参加し送信者となれること、その配送網が他のアプリケーションに対して透過性をもつこと、である。本研究では、ユーザによる世界規模での自由な1対多型通信を実現するためのアーキテクチャとしてupamulを提案する。upamulはユーザを収容するエッジネットワークにIPマルチキャストを用いることによって、他のアプリケーションに対する透過性を実現し、配送網を

リングトポロジーのオーバーレイで構築することによって、ユーザが自由に1対多型の通信を行える配送網を少ない計算量で構築する。

4 関連研究

既存の1対多型の通信を実現するためのアーキテクチャとして、IP層で実現するアーキテクチャと、オーバーレイを用いて実現するアーキテクチャがある。さらに、オーバーレイを用いるアーキテクチャは、網構築の手法によって以下の3種類のモデルに分類することができる[3]。既存のIPマルチキャストを使用可能なネットワーク間をオーバーレイで繋ぐハイブリッドマルチキャスト、エンドホストがオーバーレイの配送網を構築するアプリケーションレイヤーマルチキャスト(ALM)、ネットワーク上に設置されたプロキシルータによってオーバーレイの配送網を構築するオーバーレイマルチキャストである。

IP層で1対多型の通信を実現する手法としてXcast[4]がある。Xcastでは、送信者は全てのXcastデータの受信者のアドレスをリストとしてヘッダに付与しパケットを送信する。Xcastパケットを受信したXcastルータは、まず各受信者アドレスごとに通常のユニキャストの経路表を検索する。そしてネクストホップが同一である受信者アドレスをリストにして1つのXcastパケットを作成し、ネクストホップへ転送する。このように、Xcastはヘッダに受信者のアドレスをリストとしてヘッダに付加し、分岐点でコピーすることによって1対多型の通信を実現している。Xcastでは、宛先アドレスごとのネクストホップ検索は通常のユニキャストの経路表によって行われるため、マルチキャスト用に新たにトポロジーを構築する必要が無いという利点がある。しかし、送信者が全ての宛先をヘッダに付加する以上、パケット長は有限であるため、1つのグループに許容できる受信者数には限界がある。

一方、ハイブリッドマルチキャストモデルである Beichuan Zhang らの研究 [5] では、既存の IP マルチキャストを利用可能なネットワークをオーバーレイの配送網を用いて相互に接続する、という手法を提案している。また、別の手法として Automatic Multicast without Explicit Tunnels(AMT)[6]がある。AMTでは、既存のIPマルチキャスト網に設置されたAMT-Relayと、非IPマルチキャスト網内のネットワークに設置されたAMT-Gatewayの間で、IPマルチキャストパケットをカプセル化し転送することでIPマルチキャスト到達範囲を拡大する。しかし、

このモデルにも問題がある。オーバーレイで接続される既存のIPマルチキャスト網が、運用者によって送信者を静的に設定するように設計されていた場合、ユーザが自由にIPマルチキャストの送信者となることはできない。このように、ハイブリッドマルチキャストは、既存のIPマルチキャスト網の中でユーザが自由にIPマルチキャストの送信者になることはできないという問題を解決できていない。

ALMはエンドホストのみでオーバーレイの配送網を構築する。これにより、バックボーンネットワークの構成に影響されずに配送網を構築することができる。Multicast Middleware(MM)[7]では、エンドホストにインストールされるアプリケーションであるMMがオーバーレイの配送網を構築する。拡張性のあるオーバーレイによる配送網に加え、MMは配送する実トラフィックにIPマルチキャストを用いることで網のアプリケーション透過性を実現している。アプリケーションは、MMによって作成された仮想インターフェースを経由することでIPマルチキャストパケットの送受信を行うことができる。このように、ALMのモデルでは、エンドホストのアプリケーションが配送網を構築する。そのため、網を拡大するにはエンドホストに変更を加える必要がある、構築される網は構築するアプリケーションに依存してしまう、といった問題点がある。結果として、ALMには、構築される配送網に他のアプリケーションに対する透過性がないという問題がある。

エンドホストが配送網を構築するALMに対して、オーバーレイマルチキャストではネットワーク上に設置されたプロキシルータが配送網を構築する。エンドホストはプロキシルータからユニキャストまたはIPマルチキャストでトラフィックを受信する。オーバーレイマルチキャストでは、計算資源やネットワーク資源の乏しいことが予想されるエンドホストを配送網から除外することで高いパフォーマンスを実現している。ScatterCast[8]ではさらに、プロキシルータにモジュールを追加可能にすることで、オーバーレイ上でのトランスポート層に拡張性を持たせている。しかし、配送網はパスベクター型のルーティングプロトコルによって木構造として構築される。そのため、エンドホストの参加や離脱が頻繁に発生する場合には配送網トポロジーの計算量が大きくなる。このように、既存のオーバーレイマルチキャストのモデルは、配送網を送信者を根とする木構造で構築するものである。そのため、大規模な網でノードの参加や離脱が多くなると計算量が増大し、ルータの負荷が高くなる。これにより、オーバーレイマルチキャストを世界規模で実効的に動作させることは困難である。

5 設計

世界規模での自由な1対多型通信を実現するための機能要件を以下に示す。

- ルータの参加時、離脱時の計算量が少なく、複数の送信者を許容できる配送網の構築手法
- 配送網を構築する際の宛先発見
- 送信者から複数の受信者へのデータ配送

ユーザが世界規模で自由に1対多型の通信を行うためには、配送網は、ルータの参加と離脱が頻繁に起きる場合、送信者が複数存在する場合などに計算量を抑える必要がある。そこで、upamulは配送網をリングトポロジで構築する。各ルータはリングにおける数ホップ先のルータまでのアドレスを記憶し、これを用いてネクストホップのルータが離脱した際の復旧処理を離脱を検知したルータのみが行う。これにより、ルータ離脱時のトポロジ修復を局所化し、計算量を減らすことができる。また、ユーザの送信したデータは、この配送網を通じて受信者へと配送される。その際、適切に受信者全てにデータが配送されること、配送網内でパケットがループするのを防止することが必要である。upamulは網のアプリケーション透過性の実現のため、ユーザを収容するエッジネットワークにIPマルチキャストを用いる。その際、IPマルチキャストはIGMPによってホストのグループ管理を行うため、オーバーレイネットワークへIPマルチキャストグループの状態を通知するために、upamulはサブネットに対してIGMP Querierとして動作する必要がある。さらに、IGMPをトリガーにルータが新たにグループの配送網に参加する際に、参加するルータが参加要求を送るべき宛先を発見する仕組みが必要である。

5.1 モデルの概要

upamulはユーザを収容するエッジネットワークにIPマルチキャストを、配送網にオーバーレイネットワークを用いるオーバーレイマルチキャストモデルのネットワークアーキテクチャである。配送網を構築するupamulルータは、サブネットに1つずつ設置され、サブネットに対してIPマルチキャストルータとして動作する。upamulルータがサブネットから受信したIPマルチキャストパケットは、カプセル化され、オーバーレイで構築されたリングトポロジの配送網を通じて配送されていく。

現行の多くのホストはIPマルチキャストをIP層の機能として既にサポートしている。そのため、エッジネットワークにIPマルチキャストを用いることで多くのホストが変更を必要とすることなくupamulの網を利用することができる。またIPマルチキャストはIP層の機能であるため、上位層のアプリケーションに対して透過性を持つ。これにより、アプリケーションはupamulの網を用いて様々なサービスを展開することができる。

upamulは配送網をオーバーレイで構築する。これにより、バックボーンネットワークの構成に影響されることなく、配送網を拡張することができる。upamulは用途に合わせて以下に示す2種類のオーバーレイネットワークを構築する。その概要を図1に示す。

- All-Node Channel : upamul網に参加する全てのルータによって構築されるオーバーレイネットワークであり、Group Channelに参加するルータ群を取得するためのチャンネル。
- Group Channel : グループアドレスに参加しているルータによって構成されるリングトポロジのオーバーレイネットワークであり、マルチキャストデータの配送を行うためのチャンネル。

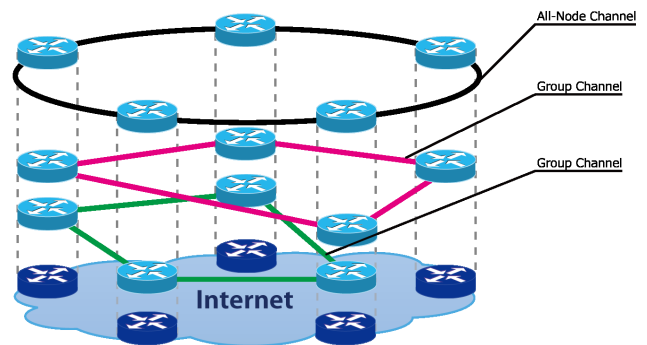


図1: オーバーレイネットワークの概要

All-Node Channelは、あるルータが新しくGroup Channelに参加する際の宛先発見に利用される。宛先発見時の高速な検索と負荷分散を実現するため、All-Node ChannelはDHTで構築される。一方Group Channelはマルチキャストデータの配送網として利用される。配送網構築時の計算量を軽減し、ルータ離脱時のトポロジ変動を小規模に抑えるため、Group Channelはリングトポロジで構成される。これにより、送信者ごとに配送網を構築する必要がなく、送信者数が増加してもルータの計算量が増大する

ことはない。また、Group Channel をグループアドレスごとに構築することによって、あるグループアドレスのトラフィックの影響をその Group Channel だけに限定し、グループに参加していないルータへの余剰なトラフィックを削減することができる。Group Channel の概要を図 2 に示す。

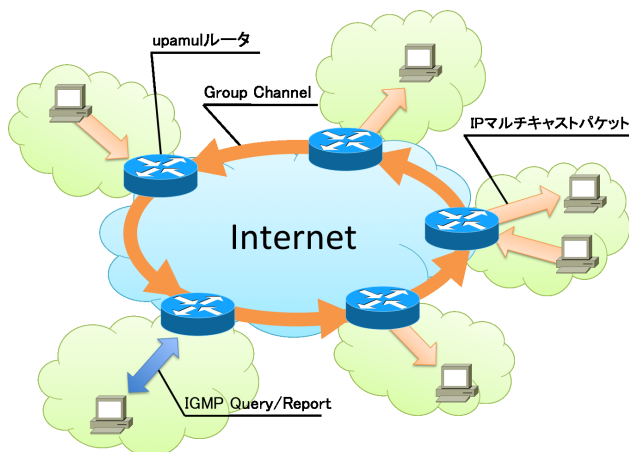


図 2: Group Channel の概要

1つのマルチキャストグループアドレスについての Group Channel は、図 2 のように構成される。ユーザを収容するサブネットに設置された upamul ルータは、ホストから IGMP Report を受け取り、指定された Group Channel に参加する。また、エンドホストとの IGMP メッセージのやり取りによって Group Channel への参加状態を維持する。Group Channel はそのグループのマルチキャストグループアドレスに参加する upamul ルータによって、リングトポロジーのオーバーレイで構築される。エンドホストの送信した IP マルチキャストパケットは、まず upamul ルータに受信され、カプセル化されて Group Channel におけるネクストホップへと転送される。Group Channel を通じてカプセル化されたパケットを受け取った upamul ルータは、カプセル化を解き、受信ノードが接続するサブネットへと送信する。そしてカプセル化されたパケットをネクストホップへと転送する。

upamul は、参加要求を行う際に宛先を発見する Discovery, Group Channel を構築, 維持, 復旧するためのプロトコルである Join, Keep Alive, Repair, IP マルチキャストデータの配送を行う Forwarding, ホストのグループ制御を行うための IGMP Querier によって構成される。

5.2 Discovery

upamul ルータ (以下ルータ) が新規に Group Channel に参加する際、参加要求を送るべき相手を All-Node Channel を用いて発見する。All-Node Channel は upamul 網に参加する全ての upamul ルータによって構築される。All-Node Channel は DHT によるオーバーレイネットワークで構築され、マルチキャストグループアドレスを key として、そのグループアドレスの Group Channel に参加しているルータのアドレスが value として格納される。あるルータが新しい Group Channel に参加する際、参加要求を送る相手を発見するために All-Node Channel を利用する。ルータの Group Channel への参加は 5.3.1 章に詳述する。

本論文では、All-Node Channel を構築する DHT のアルゴリズムに Chord を、実装には Overlay Weaver[9] を用いた。

5.3 Group Channel Management

Group Channel の管理のために、以下に示す Join, Keep Alive, Repair の 3つのプロトコルを用いる。

5.3.1 Join

ルータがあるグループアドレスのマルチキャストトラフィックを受信するには、そのアドレスの Group Channel に参加する必要がある。ルータがある Group Channel へ参加するには、まず All-Node Channel に対してグループアドレスを key として検索を行う。これにより最初に参加要求を送るべきルータのアドレスを取得する。その後、参加する Group Channel に既に参加しているルータが 0 台の場合、1 台の場合、2 台以上の場合で異なる処理を行う。以下では、ある Group Channel へ参加しようとしているルータをルータ A、既にその Group Channel に参加しており、ルータ A からの要求を受け取って返答するルータをルータ B とする。

まず、まだ Group Channel に参加しているルータが存在しない場合の参加処理を示す。ルータ A は参加したい Group Channel のアドレスを key として All-Node Channel で検索を行う。しかし、まだその Group Channel は存在していないので値は返らない。するとルータ A は参加する Group Channel のアドレスを key、自身のアドレスを value として All-Node Channel に追加する。そして、その Group Channel の最初の 1 台となる。

既に Group Channel に参加しているルータが1台の場合の参加処理を示す。ルータ A は参加したい Group Channel のアドレスを key として All-Node Channel で検索を行い、返ってきたルータ B のアドレスに対して参加要求を送る。参加要求を受け取ったルータ B は、自身のアドレスをネクストホップとるようにルータ A に返答する。またルータ B はその Group Channel におけるネクストホップをルータ A に設定する。返答を受け取ったルータ A はその Group Channel におけるネクストホップをルータ B に設定する。最後に、ルータ A は All-Node Channel に対して、参加した Group Channel のアドレスを key に自身のアドレスを value として追加する。

既に Group Channel に参加しているルータが2台以上存在する場合の参加処理を示す。まず、ルータ A は参加したい Group Channel のアドレスを key として All-Node Channel で検索を行う。1つの key に対して複数のアドレスが格納されている場合、ルータは格納された時期が新しいアドレスから順番に参加要求を送る。ここでは、最後に参加したルータはルータ B であるとする。ルータ A は最後に参加したルータであるルータ B に対して参加要求を送る。参加要求を受け取ったルータ B は、その Group Channel におけるルータ B のネクストホップをルータ A のネクストホップとるように返答する。そして、ルータ B は自身のネクストホップをルータ A に設定する。ルータ A は受け取ったアドレスをその Group Channel におけるネクストホップに設定する。最後に、ルータ A は All-Node Channel に対して、参加した Group Channel のアドレスを key に自身のアドレスを value として追加する。

5.3.2 Keep Alive

ルータの離脱を検知し復旧するために、各ルータは Group Channel ごとにネクストホップに対して定期的に Hello メッセージを送ることで生存確認を行う。Hello メッセージの応答には応答するルータのネクストホップのアドレスを入れて応答をする。これにより、全てのルータは2ホップ先のルータのアドレスまでを記憶する。ネクストホップが一定時間 Hello に応答せず、離脱したと判断した場合、2ホップ先のルータのアドレスを用いて復旧処理を行う。また、Hello を送り始める際に、Hello を送るルータがネクストホップのルータに対して1つ前のルータであることを登録する処理を行う。これにより、意図しない前ホップのルータが生じることを防ぐ。

5.3.3 Repair

Hello によってネクストホップのルータが離脱したことを検知した際、復旧処理を行う。復旧処理は Group Channel に参加しているルータの数が2台の場合と3台以上の場合で異なる処理を行う。以下では、離脱したルータをルータ A、ルータ A の前ホップのルータをルータ B とする。

ある Group Channel に参加しているルータが2台の場合、まずルータ B はルータ A が一定時間 Hello に応答しないことで離脱を検知する。次にルータ B は Hello によって学習した2ホップ先のルータがルータ B 自身であることを確認する。これにより、ルータ B はその Group Channel に参加しているルータがルータ B だけであることを認識する。最後に、All-Node Channel に対してその Group Channel のアドレスを key として離脱したルータ A のアドレスを削除する。

ある Group Channel に参加しているルータが3台以上の場合、まずルータ B はルータ A が Hello に応答しないことで離脱を検知する。次にルータ B は Hello によって学習した2ホップ先のルータのアドレスをネクストホップとして設定し、そのルータに対して自身が前ホップのルータであることを登録する。そして生存確認のため Hello を送信し始める。また、上記で新しくネクストホップとして設定したルータも同時に離脱していた場合、ルータ B は Hello への応答がないことでその離脱を検知し、参加処理を始めからやり直す。

5.4 Forwarding

IP マルチキャストパケットの配送は、パケットのグループアドレスの Group Channel を一周することで行われる。ユーザの送信した IP マルチキャストパケットは、まず up-amul ルータによって受信される。up-amul ルータはそのパケットをカプセル化し、そのパケットのグループアドレスに応じた Group Channel へと配送する。カプセル化する際、ヘッダにそのパケットのグループアドレス、最初に Group Channel に配送したルータのアドレスをオリジネータアドレスとして付与する。Group Channel からカプセル化されたパケットを受け取ったルータは、カプセル化を解いてユーザを収容するサブネットに流す。また、カプセル化されたものを Group Channel のネクストホップへと配送する。ルータがカプセル化されたパケットを受け取った際、ネクストホップがヘッダに記載されたオリジネータ

であった場合、そこで Group Channel への配送を止める。これによりパケットのループを防止する。

5.5 IGMP Querier

upamul はユーザを収容するエッジネットワークに IP マルチキャストを用いるため、参加すべきグループの管理を IGMP によって行う。そのために upamul ルータはエッジネットワークに対して IGMP Querier として動作する。upamul ルータはユーザの送信した IGMP Report をトリガーとして、その Group Channel への参加処理を実行する。また、IGMP をトリガーとして自身が参加する Group Channel からの離脱を行う。

6 評価

本章では、upamul ルータの転送性能の限界を測定するため、一定のトラフィック負荷をかけた際の実効帯域とパケットロス率についての評価を行う。upamul ルータはマルチキャストルータであり、1つのサブネットに1台ずつ設置される。そして、サブネットへのマルチキャストトラフィックは全て upamul ルータを経由して配送される。また、利用シナリオとして、upamul によって提供される IP マルチキャスト配送網を用いてユーザが自由にライブストリーミングを行うことを想定している。そのため upamul ルータがどの程度の帯域まで実効的に配送できるのかが重要である。そこで、本論文では upamul ルータの実効帯域についての評価を行った。評価項目として、upamul を用いて一定のマルチキャストトラフィックを配送した際のパケットロス率と実行帯域について測定し、評価を行った。

6.1 実験環境

実験環境として、upamul ルータの転送性能の限界を測定するために upamul ルータ以外の要素がボトルネックにならない環境を用意した。まず、実験環境として3つのセグメントを用意し、各セグメントに upamul ルータと測定用 PC を1台ずつ設置した。この実験環境のトポロジーを図3に示す。

図3の実験環境において、マルチキャストトラフィックの配送網として、それぞれ別のセグメントに設置された3台の upamul ルータを用いて1つの Group Channel を構成した。まず 192.168.1.0/24 のセグメントに設置された測

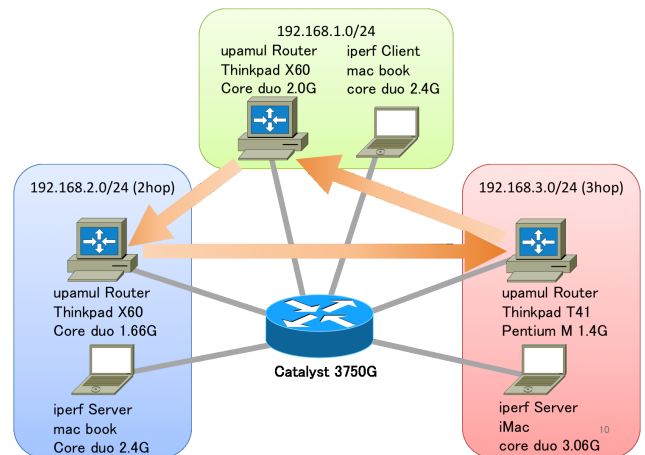


図 3: 実験トポロジー

定用 PC がマルチキャストトラフィックを生成する。このトラフィックは Group Channel を通じて、192.168.2.0/24 のセグメントに、続いて 192.168.3.0/24 のセグメントに配送される。そして、各セグメントの測定用 PC がこのマルチキャストトラフィックを受信することで、パケットロス率及び実効帯域の測定を行った。

本実験では、マルチキャストトラフィックの生成及び受信、パケットロス率と実効帯域の測定に iperf を用いた。iperf で生成したマルチキャストトラフィックはペイロードが 1470Byte の UDP パケットである。測定は、iperf Client が生成するマルチキャストトラフィックの帯域幅を 10Mbps から始め、10Mbps ずつ増加させて各帯域について 60 秒間測定することで行った。

本実験において、upamul ルータとして使用した機器を以下に示す。OS は全て Debian Lenny 5.0.5 を使用した。また、測定用 PC、セグメント間のルータを含む全機器の NIC は全て 1Gbps の全二重である。

- ThinkPad x60, CPU Intel Core 2 Duo 2.4GHz, Memory 2G, NIC 1000BASE-T
- ThinkPad x60, CPU Intel Core 2 Duo 1.66GHz, Memory 2G, NIC 1000BASE-T
- ThinkPad T41, CPU Intel Pentium M 1.44GHz, Memory 1G, NIC 1000BASE-T

6.2 結果と考察

6.1 章の実験によって計測した、トラフィック流量と実効帯域の関係を図4に、トラフィック流量とパケットロス

率の関係を図5に示す。

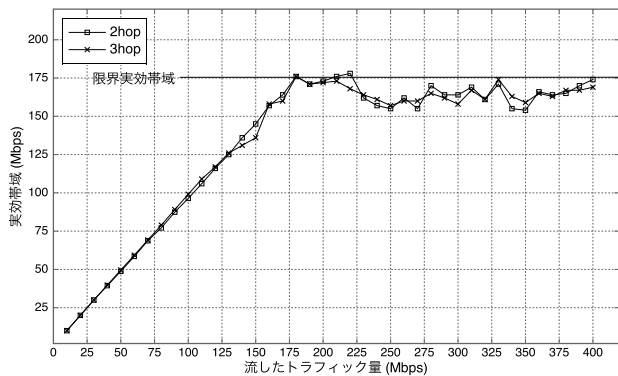


図 4: トラフィック流量と実効帯域

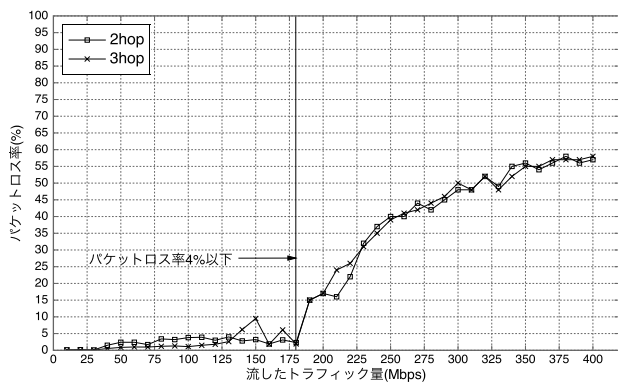


図 5: トラフィック流量とパケットロス率

図4, 図5における2hop, 3hopとは, それぞれ Group Channel 上で upamul ルータを 2hop したセグメント (192.168.2.0/24) の測定用 PC で測定された結果, 3hop したセグメント (192.168.3.0/24) の測定用 PC で測定された結果, ということである。

トラフィック流量と実効帯域のグラフ (図4) から, 現在の upamul ルータの限界実効帯域がわかる。図4において, iperf によって流したトラフィック流量が 180Mbps 以上の際は 2hop, 3hop ともに実効帯域が 175Mbps 以下で推移している。このことから, 現在の upamul ルータの転送性能の限界は 175Mbps 程度であるといえる。

トラフィック流量とパケットロス率のグラフ (図5) から, upamul ルータが現在どの程度のストリームを流すことができるかわかる。図5において, トラフィック負荷が 130Mbps 以下まではパケットロス率は 4% 以下である。動画を流す際に支障の無いパケットロス率を 4% 以下である

とすると [7], upamul ルータは合計 130Mbps までであればストリーミングを実効的に行うことができるといえる。また, トラフィック負荷が 180Mbps 以上になると, トラフィック流量の増加に伴ってパケットロス率も上昇している。これは, 転送性能の限界が 175Mbps であることを示している

7 結論と課題

本論文では, ユーザによる IP マルチキャストを用いた世界規模での自由な 1 対多型の通信を実現するアーキテクチャとして, upamul の提案を行った。upamul はエッジに IP マルチキャストを用いることによって網のアプリケーション透過性を実現し, 配送網をリングトポロジーのオーバーレイネットワークで構築することによってユーザが自由に IP マルチキャストを送受信できる環境を提供する。

本論文では, 小規模な実験環境において upamul の実効帯域についての評価を行い, upamul の限界実効帯域が 175Mbps であることを示した。また, 合計が 130Mbps 以下であれば, パケットロス率を 4% 以下に抑えられることを示した。これにより, トラフィックの合計が 130Mbps 以下であれば, ストリーミングを実効的に行えることがわかった。

課題として, 本論文では行わなかった広域での実効帯域に関する評価, 配送遅延に関する評価, 複数ストリームを流した際の評価, 配送網を構成するプロトコルの評価を行う必要がある。また, 配送網をリングトポロジーで構築するため, 1つのグループへの参加ルータが増えるとジッターの増大や特定のルータ間のリンクがボトルネックになってしまうといった問題が予想される。そのため, 実ネットワークを考慮したオーバーレイでのトポロジー構築が必要であると考えられる。

参考文献

- [1] A. Adams, J. Nicholas, and W. Siadak. Protocol Independent Multicast - Dense mode (PIM-DM): Protocol Specification (Revised). RFC 3973, IETF, Jan 2005.
- [2] B. Fenner, M. Handley, H. Holbrook, and I. Kouvelas. Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification (Rivised). RFC 4601, IETF, August 2006.
- [3] John Buford. Survey of ALM, OM, Hybrid Technologies. IETF SAMRG, July 2006.

- [4] R. Boivie, N. Feldman, IBM, Y.Imai, WIDE / Fujitsu, W. Livens, ESCAUX, D. Ooms, and OneSparrow. Explicit Multicast (Xcast) Concepts and Options. RFC 5058, IETF, Nov 2007.
- [5] Beichuan Zhang, Wenjie Wang, Sugih Jamin, Daniel Massey, and Lixia Zhang. Universal IP Multicast Delivery. *NGC*, Oct 2002.
- [6] D. Thaler, M. Talwar, A. Aggarwal, Microsoft Corporation, L. Vicisano, Qualcomm Inc., and T. Pusateri. draft-ietf-mboned-auto-multicast-10. *IETF*, Mar 2010.
- [7] Marc Brogle, Dragan Milic, and Torsten Braun. Supporting IP Multicast Streaming Using Overlay Networks. *ACM*, August 2007.
- [8] Yatin Chawathe. Scattercast: An Adaptable Broadcast Distribution Framework. *ACM Multimedia Systems Journal on Multimedia Distribution*, Oct 2002.
- [9] 首藤一幸 田中良夫 関口智嗣. オーバレイ構築ツールキット overlay weaver. **先進的計算基盤システムシンポジウム (SACISIS2006)**, May 2006.