



Title	Flexcastによる段階的導入に優れたマルチキャストシステムの設計と実装 : システム開発論文特集
Author(s)	井上, 武; 谷, 誠一郎; 高橋, 宏和 他
Citation	電子情報通信学会論文誌. D-I, 情報・システム, I-情報処理, J88(2), 272-291
Issue Date	2005-02-01
Doc URL	https://hdl.handle.net/2115/47395
Rights	copyright©2005 IEICE
Type	journal article
File Information	40_ieice_2005.pdf



Flexcast による段階的導入に優れたマルチキャストシステムの設計と実装

井上 武^{†a)} 谷 誠一郎^{††} 高橋 宏和[†] 湊 真一^{†*}
宮崎 敏明[†] 豊島 鑑[†]

Design and Implementation of the Incrementally Deployable Multicast System Based on Flexcast

Takeru INOUE^{†a)}, Seiichiro TANI^{††}, Hirokazu TAKAHASHI[†], Shin-ichi MINATO^{†*},
Toshiaki MIYAZAKI[†], and Kan TOYOSHIMA[†]

あらまし IP multicast は、これまで 10 年以上にわたって研究開発が進められてきたが、インターネット全域に広く展開されるには至っていない。この原因の一つとして、IP multicast を使ったマルチキャストシステムは、初期投資が大きいという点が考えられる。Flexcast は IP multicast と同様にルータで分岐を行う方式であるが、マルチキャスト機能をネットワーク層からより上位の層へと移したため、部分的な導入でサービスを開始できるという特徴をもつ。我々はこのような Flexcast の特徴を使って、少ない設備投資でサービスを開始でき、需要に合わせて拡張可能なマルチキャストシステムを開発した。Flexcast ルータの開発にあたっては、WWW 検索システムや GRID computing で大きな成果を挙げている PC クラスタを応用し、初期投資の抑制と高い拡張性を実現するとともに、ネットワーク装置に求められる高い可用性も実現した。グループ管理システムの開発では、IP multicast との互換性を維持し、既存アプリケーションの利用を可能とすることで、開発コストを抑制した。更に、IP multicast において長年の懸案となっていたグループアドレス管理に関するスケーラビリティ問題を、管理権限を局所化することにより解決した。最後に、市販 PC に実装して評価実験を行い、本提案システムが Gbit/s レベルの転送能力を有しており、また、高い可用性を備えることを明らかにした。

キーワード マルチキャスト, マルチキャストルータ, グループ管理, 段階的展開, Flexcast

1. ま え が き

インターネットが広く普及し、現在では誰もが気軽に WWW (World Wide Web) や電子メールを楽しめるようになった。これまでは、文字や静止画のような小容量コンテンツを中心に利用されてきたインターネットであるが、ADSL や FTTH によりアクセスラインの広帯域化が進むにつれ、映像のような広帯域コンテンツに対する需要が増加している。しかし、現在

のインターネットはユニキャストによる配信が一般的であり、受信者数に比例してトラフィックが増大するという課題を抱えている。この課題を解決するために、10 年以上にわたってマルチキャスト配信技術の研究が進められてきた。一般にマルチキャストでは、送信者を根、受信者を葉とする木状の配信経路を構築する。送信されたパケットは、節に位置するルータによって複製され、各受信者まで届けられる。マルチキャストを導入することによって、トラフィックを大幅に削減できることが分かっている [11], [32]。

このように大きな利点をもつマルチキャストであるが、その長い歴史にもかかわらず、MBone [17] のような仮想実験網や企業内、学内に閉じて利用されるにとどまっている。これにはいくつかの原因が考えられるが [16]、我々は以下の問題点に注目した。IETF 標準として採用され、最も広く知られているマルチキャ

[†] 日本電信電話株式会社 NTT 未来ねっと研究所, 横須賀市
NTT Network Innovation Laboratories, NTT Corporation,
1-1 Hikari-no-oka, Yokosuka-shi, 239-0847 Japan

^{††} 日本電信電話株式会社 NTT コミュニケーション科学基礎研究所,
厚木市

NTT Communication Science Laboratories, NTT Corpora-
tion, 3-1 Morinosato Wakamiya, Atsugi-shi, 243-0198 Japan

* 現在, 北海道大学大学院情報科学研究科

a) E-mail: takeru.inoue@m.ieice.org

スト方式に IP multicast [13] がある。しかし、マルチキャスト機能をネットワーク層で実現しているため、経路上の全ルータが対応するまでマルチキャストサービスを開始できない。また、実際のサービスでは、広帯域コンテンツを中心に利用されると予想されるため、ルータには Gbit/s レベルの高い転送能力が求められる。ところが、サービス開始前に、高性能マルチキャストルータを展開、運用することは、経済的リスクが大きく、現実的ではない。この問題を解決するために、少ない設備投資でサービスを開始し、需要の拡大に合わせて段階的に拡張可能であるマルチキャストシステムの登場が望まれている。

我々は、上記の課題を解決する新たなマルチキャスト方式、Flexcast の提案を行ってきた [26], [31], [34]。Flexcast は、より上位の層でマルチキャスト機能を実現しており、ルータ間の通信はユニキャストによって行われる。このため、ネットワーク全体への展開を待たずして、サービスを開始することができる。本論文は、Flexcast を用いたマルチキャストシステムの開発について論じる。我々は、少ない設備投資でサービス開始可能であること、需要に合わせて拡張可能であること、をマルチキャストシステムに対する課題ととらえ、開発を行った。また、ネットワークシステムに必須とされる可用性も課題に含め、検討を進めた。開発システムは、Flexcast ルータとグループ管理システムに大別される。Flexcast ルータの開発にあたっては、GoogleTM [39] に代表される WWW 検索システムや GRID computing [40] で大きな成果を挙げているクラスタシステムを応用し、小規模 PC クラスタによってルータシステムを構成した。グループ管理システムの開発では、IP multicast との互換性やスケーラビリティに留意した。

以下、2. で Flexcast を始めとしたマルチキャスト方式を振り返る。3. で開発システムへの要求条件を確認し、4., 5. で Flexcast ルータとグループ管理システムの構成方法を提案する。評価結果を 6. で述べる。7. では関連技術について触れ、8. で本論文をまとめる。

2. マルチキャスト方式

2.1 マルチキャスト方式比較

本章では、現在までに提案されているマルチキャスト方式を 4 種類に分類し、それぞれの得失を明らかにする。比較にあたっては、段階的展開 (incremental

deployment) のサポート、配信経路 (delivery path)、受信者数 (number of recipients) に焦点を当てる。

IP multicast [13] は、IETF 標準として採用されているマルチキャスト方式である。その特徴は、クライアント管理と経路制御に異なるプロトコルを用いる点、マルチキャスト機能をネットワーク層で実現している点にある。PIM-SM (protocol independent multicast-sparse mode) [14] に代表される経路制御プロトコルは、送信者を根、受信者を葉とする最短経路木^(注1)を構築し、配信を行う。クライアント管理プロトコルには、IGMP (Internet group management protocol) [8] あるいは MLD (multicast listener discovery) [36] を用いる。IGMP/MLD については、2.2 で説明する。一方、マルチキャスト機能をネットワーク層で実現するために、IP multicast ではグループアドレスという特殊な IP アドレスを用いている。ルータは、ユニキャストとは異なるメカニズムによってグループアドレスあてのパケットを転送する。しかし、このメカニズムを実装しないルータが配信経路上に存在すると、そこでパケットは廃棄されてしまう。結局、IP multicast は、対応ルータがネットワーク全体に展開されないうちは満足に動作しないという課題を抱えることとなった^(注2)。

IP multicast の対極に位置する方式が、End-system multicast [1], [10] である。End-system multicast は、マルチキャスト機能を上記層に実装する。そして、各受信者を節とする配信経路を構築し、ユニキャストで節間を通信する。このため、ルータへの機能追加は不要であり、受信者にソフトウェアを配布することで迅速にサービスを開始することができる。しかし、パケットの複製を受信者が行うため、配信経路は大きく迂回することとなり、IP multicast ほどの大きなトラフィック削減効果は期待できない。また、品質やセキュリティを他の受信者に負うという課題も残される。

Xcast [3] は、パケットヘッダに受信者アドレスを列挙し、配信経路上のルータが必要に応じてパケットを

(注1)：本論文では、各受信者から送信者への最短経路によって構成される木を指す。

(注2)：過渡的な対処法として、対応ルータ間にトンネルを設定し、非対応ルータを越えて配信を行うことができるが、運用コストの増加という課題がある [2]。運用コストを削減するために、受信者から遠隔の IP multicast ルータへと自動的にトンネルを構築する方式も提案されている [20], [35]。しかし、このようにして構築されるトンネルは、マルチキャスト特有の木状経路ではなく、ユニキャストのように放射状となる。このため、IP multicast の利点であるトラフィック削減効果が薄れてしまうという欠点がある。

表 1 マルチキャスト方式比較
Table 1 Comparison of some multicast methods.

Multicast method	Incremental deployment	Delivery path	# of recipients
IP multicast	Not supported	Shortest path	Unlimited
End-system multicast	Supported	Redundant path	Unlimited
Xcast	Supported	Shortest path	Restricted by packet length
Flexcast	Supported	Shortest path	Unlimited

複製しながら配信を行う方式である。ユニキャストアドレスをあて先とするため、配信経路上に非対応ルータが存在してもパケットが廃棄されることはない。しかし、受信者数に比例してパケットヘッダが大きくなるため、配信数に限界が生じる。また、パケットを転送するたびに、列挙されたすべてのアドレスに対して経路探索を行うため、ルータの転送負荷が非常に高くなる。複数パケットにわたって Xcast を行うことで、配信数の限界を解消する改良方式 [5] も提案されているが、長すぎる受信者リストは、より大きなオーバーヘッドをもたらす。

Flexcast [26], [31], [34] は、上記 3 方式それぞれの特徴を併せもつマルチキャスト方式である。ここでは特徴を述べ、詳細は次節で説明する。Flexcast は、End-system multicast や Xcast と同様に、マルチキャスト機能をより上位の層に実装し、ユニキャストアドレスにより通信を行う。このため、Flexcast を実装しないルータが混在したネットワークであっても、マルチキャスト配信を行うことができる。Flexcast の配信木は、クライアントからサーバに至るユニキャスト経路を束ねた最短経路木になるため、End-system multicast のように経路が冗長になることはない。経路上の Flexcast ルータで分岐が行われ、その割合が増えるほどトラヒック削減効果は大きくなる。すべてのルータが対応すると、IP multicast と同等のトラヒック削減効果が得られる^(注3)。

以上の議論により、我々は Flexcast を選択した。表 1 に各方式の特徴を簡単にまとめる。

2.2 Flexcast

本節では、Flexcast について説明を行う。Flexcast の役割は経路制御とデータ配信である。信頼性やセキュリティについては、他の方式と組み合わせることで実現する。詳細は 7. で述べる。

Flexcast では、送信者（サーバ）を根とする木状の経路を構築し、配信を行う。以下、この木状の配信経路を配信木と呼ぶ。配信木は、SSM (source specific multicast) [25] と同様に、サーバアドレス S とグルー

プアドレス G のペア (S, G) によって区別される。以下、このペアを SSM チャンネル、あるいは単にチャンネルと呼ぶ。また、説明の便宜上、Flexcast ルータを次のように区別する。サーバと同一セグメントに接続されたルータを Root ルータ、クライアントと同一セグメントに接続されたルータを Leaf ルータ、その他のルータを Branching ルータとする。

Flexcast では、IP multicast と同様に、IGMP/MLD によってクライアントを管理する。このため、IP multicast 用に開発されたクライアントをそのまま利用することができる。IGMP/MLD にはいくつかのバージョンがあるが、チャンネルの特定が可能な IGMP version 3, MLD version 2 (以下、IGMPv3/MLDv2) を用いる。

続いて、Flexcast の動作概要を説明する。クライアントは、受信を希望するチャンネルを記載した IGMP membership report あるいは MLD multicast listener report を送出する。Leaf ルータは Report を受信すると、Join と呼ばれるパケットを作成し、定期的に送信する。Join パケットには受信希望チャンネルを記載し、サーバのユニキャスト IP アドレスをあて先に設定する。Join パケットは通常のユニキャスト IP ルーティングに従ってサーバへと転送される。この転送経路上に Branching ルータが設置されていると、その Branching ルータは Join パケットの送信元アドレスを Flexcast 転送表 (Flexcast forwarding table) と呼ばれる表に登録する。転送表には、チャンネルごとに要求元アドレスが記録される。そして、Leaf ルータと同様に Join パケットを作成し、サーバへと定期的に送信する。Root ルータも、Branching ルータと同様に転送表を管理する。最終的に Join パケットが Root ルータまで到達すると、配信木が完成する。

(注3) : Flexcast と同様の特徴をもつ方式に、REUNITE (REcursive UNicast TreE) [33] や HBH (Hop By Hop multicast) [12] がある。しかし、REUNITE は特殊なアルゴリズムにより配信経路を構築するため、受信者離脱時の経路再構成処理が配信木全体に波及するという課題がある [12]。また、HBH はプロトコルが非常に複雑であり、配信されない受信者が現れる可能性が指摘されている [4]。

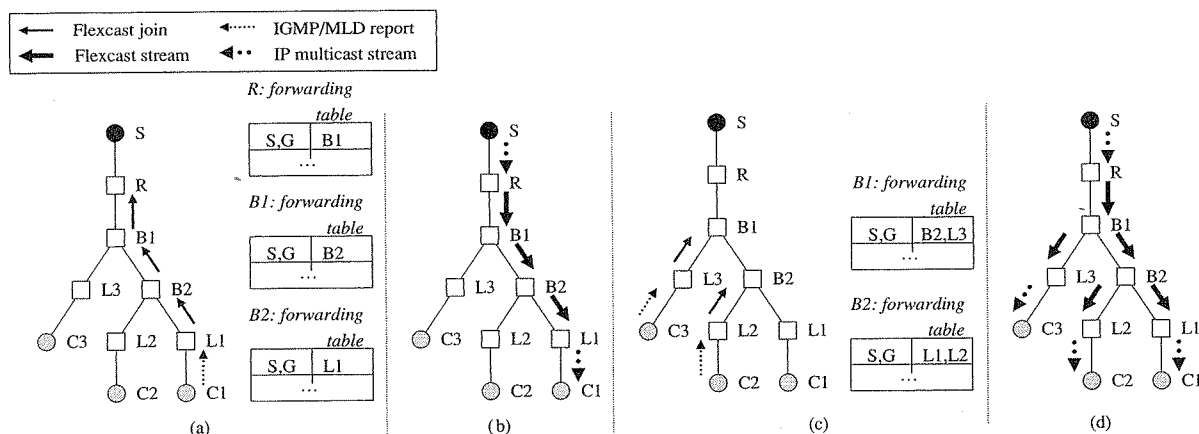


図 1 Flexcast 動作例
Fig. 1 An example of Flexcast.

Root ルータは、サーバから IP multicast パケットを受信し、Stream と呼ばれるパケットにカプセル化する。そして、転送表に登録されている要求元へとユニキャストで送信する。Branching ルータは、受信した Stream パケットを、転送表の各要求元へと転送する。最終的に Stream パケットが Leaf ルータまで到着すると、Leaf ルータによってカプセル化されていた IP multicast パケットが取り出され、クライアントへと送出される。

クライアントが受信を停止するときは、次のような動作が行われる。まず、クライアントが IGMP/MLD leave を送出する。Leaf ルータは、他に同一チャネルを受信しているクライアントが存在しないことを確認すると、サーバあてに Prune パケットを送信する。Prune パケットには、停止チャネルが記載される。Prune パケットが Branching ルータによって受信されると、Branching ルータは送信元を転送表から削除する。そして、そのチャネルの要求元がいなくなったら、Prune パケットをサーバへと送信する。Root ルータも Branching ルータと同様に、Prune パケットの送信元を転送表から削除する。要求元がいなくなったチャネルは、配信を停止する。

図 1 は、3 台のクライアント C1, C2, C3 が、チャネル (S, G) を順に要求する様子を表している。配信木の右側には、各ルータが管理する Flexcast 転送表を記す。この図を用いて、具体的な動作例を説明する。Leaf ルータ L1 は、クライアント C1 が送出した Report を受信すると、サーバ S あてに Join パケットを送信する。L1 から S への経路上に設置された Branching ルータ B2 は、Join パケットの送信元である L1 を転送表に登録し、S あてに Join パケットを送信する。同様に、

Branching ルータ B1 の転送表には B2 が、Root ルータ R の転送表には B1 が登録される (図 1(a))。R は、S が送出している IP multicast パケットを Stream パケットにカプセル化し、B1 へと送信する。Stream パケットは B2 を経由して L1 に届けられ、カプセル化を解かれた IP multicast パケットが C1 へと到着する (図 1(b))。

続いて、クライアント C2, C3 が Report を送信する。この Report を受信した Leaf ルータ L2, L3 は、L1 と同様に Join パケットを送信し、それぞれ B2, B1 の転送表に登録される (図 1(c))。R は、S が送出している IP multicast パケットを Stream パケットにカプセル化し、B1 へと送信する。Stream パケットは B1, B2 で複製され、L1, L2, L3 まで届けられる。L1, L2, L3 はカプセル化を解き、C1, C2, C3 へと IP multicast パケットを送信する (図 1(d))。

ここで、B2 が Flexcast に対応していない場合を考える。L1, L2 が送信した Join は、B2 ではなく B1 によって受信される。そして、B1 の転送表に L1, L2 が登録される。Stream パケットは、B1 で三つに複製され、L1~L3 にあてて送られる。この後、B2 を Flexcast 対応ルータに置き換えると、L1 と L2 への分岐が B2 で行われることになる。つまり、最初からすべてのルータを Flexcast 対応にしておく必要はなく、後から順次 Flexcast 対応ルータを増やしていくことが可能であり、これによって配信効率を向上させることができる。Flexcast ルータの割合が増えるほどトラヒック削減効果は大きくなり、全ルータが対応すると IP multicast と同じ配信木を構築することができる。

ところで、本節では IGMPv3/MLDv2 の利用を前提として、Flexcast の説明を行った。しかし、

現在は IGMPv3/MLDv2 への移行過渡期であり、IGMPv1,2/MLDv1 を実装したクライアントもまだ多く存在する。IGMPv1,2/MLDv1 はグループアドレスのみを指定するプロトコルであり、SSM チャンネルを特定することができない。IGMPv1,2/MLDv1 クライアントに対して配信を行うためには、グループアドレスにサーバアドレスを追加し、チャンネルを取得するメカニズムが必要とされる。また、IGMPv1,2/MLDv1 を用いる場合には、グループアドレスをネットワーク全体で一意的に保たなければならないが、スケーラビリティのあるグループアドレス管理方式は提案されていない [16]。

3. マルチキャストシステムへの要求条件

本節では、マルチキャストシステムが実現すべき課題を確認する。

- 設備投資の抑制

既存資産を有効に利用した開発を行い、設備投資を抑える。例えば、Flexcast ルータの開発では、既に運用されているルータやスイッチを取り込んでシステム設計を行うことが望ましい。また、これまでに開発された IGMPv1,2/MLDv1 クライアントを利用するために、グループ管理システムによって IGMPv1,2/MLDv1 への後方互換性を実現する。

- 高いスケーラビリティ

マルチキャストシステムは、多くのユーザに対してサービスを提供することを期待されているため、高いスケーラビリティが要求される。そのため、Flexcast ルータはトラヒックの伸びに合わせて転送能力を段階的に拡張することができ、Gbit/s レベルの高い転送能力を達成することが望ましい。また、グループ管理システムは、グループアドレス管理にまつわるスケーラビリティ問題を解決することが期待される。

- 高い可用性

可用性を高めるためには、システム全体が多重化され、単一故障点が存在しないことを要求される。システムは自動的に障害を検知し、予備システムへと切り換わる。切換時間は、RIP [28]、OSPF [29] といった経路制御プロトコルと同程度の数～数十秒を目指す。

4. Flexcast ルータの設計

4.1 プロトコル設計

本節では、2.2 で紹介したアルゴリズムをプロトコルとして定義する。

2.2 で見たように、Flexcast では TCP や UDP といった下位層を規定していない。そこで、まず、下位層についての議論から開始する。1. でも述べたように、我々は主なコンテンツとして映像を検討している。映像は一般に容量が大きい、ファイル配信のように完全な信頼性は要求されないことが多い。このことから、信頼性は低い、オーバーヘッドの小さい UDP 上にプロトコルを実装する。IP 上に新たなトランスポート層を定義することも可能であるが、ファイアウォールやネットワーク監視技術との親和性が低くなるため採用しない。

Join, Prune パケットのあて先ポート番号は、あらかじめ全 Flexcast ルータで統一しておく。Flexcast ルータは、通過する UDP パケットのあて先ポート番号を検査し、Join, Prune パケットをえり分ける。Flexcast 転送表に Join パケットの送信元を登録する際には、IP アドレスに加えて、Join パケットの送信元ポート番号を記録しておく。このポート番号は、Stream パケットのあて先ポート番号となる。なお、各パケットの送信元ポート番号は任意とする。

UDP ヘッダの次に、バージョンやチャンネルといったフィールドをもつ Flexcast ヘッダを定義する。Flexcast ヘッダは、IPv4 のとき 16 バイト、IPv6 では 40 バイトとなる。具体的なヘッダフォーマットは省略する。Stream パケットでは、Flexcast ヘッダに続いて IP multicast パケットをカプセル化する。カプセル化によるオーバーヘッドは、IPv4 の場合で 44 バイトとなる^(注4)。カプセル化される IP multicast パケットが MPEG-2 Transport Stream で標準的に用いられる 1,356 バイトであるとする、カプセル化によるオーバーヘッドは約 3.2% のオーバーヘッドとなる。

4.2 クラスタシステム

一般的に、マルチキャストのように基本的なネットワーク機能は、ルータあるいはスイッチと呼ばれるネットワーク専用装置に実装される。複数ベンダの装置によって構成されているネットワークに新しい機能を追加する場合は、ベンダごとに開発を行い、相互運用性の検証作業が必要となる。また、広帯域トラヒックのリアルタイム複製処理を伴うマルチキャスト機能は、通常のネットワーク装置で用いられる CPU では処理能力が不足し、専用のハードウェアが必要となる場合もある。結果的に、マルチキャストルータの開発

(注4) : 20 (IP ヘッダ) + 8 (UDP ヘッダ) + 16 (Flexcast ヘッダ) = 44.

は高くつくことが多い。

一方、オフィスや家庭向けコンピュータである PC は、低価格で処理能力の高い CPU を備えている。比較的負荷の高い機能であっても、ソフトウェア実装で処理することができる。更に、近年では、GoogleTM [39] に代表される WWW 検索システムや GRID computing [40] において、PC で構成したクラスタシステムを用いて高い処理能力と可用性を実現することに成功している。クラスタシステムとは、複数のコンピュータを組み合わせる一つの機能を実現するシステム形態のことであり、安価なコンピュータを用いて高い処理能力と可用性を得ることができる。

我々は、運用中のルータあるいはスイッチに小規模な PC クラスタを接続し、システムとして Flexcast ルータ機能を実現するアプローチを選択する。既存設備を有効利用し、また開発コストや装置コストを抑えることで、初期投資を抑える。更に、クラスタシステムを採用することにより、高いスケーラビリティと可用性を期待することができる。Flexcast 以外のパケットは既存のルータあるいはスイッチで処理されるため、従来どおりの信頼性が確保される。

以下、ルータあるいはスイッチと PC クラスタをまとめて Flexcast ルータシステムと呼ぶ。次節より、Flexcast ルータシステムの構成方法を具体的に説明する^(注5)。

4.3 Branching ルータシステム

4.3.1 概要

本節では Branching ルータシステムの構成方法を提案する。図 2 に構成例を示す。図に示すように、既に稼働中のルータあるいはスイッチに PC クラスタを接続する。以降、PC クラスタが接続されるルータあるいはスイッチを、母ルータ (mother router) と呼ぶ。PC クラスタは、負荷分散 (load balancer) 機能をもつ PC と、Branching ルータ機能をもつ PC によって構成される。前者を Balancer PC、後者を Branching PC と呼ぶ。

Branching PC は、2.2 で述べた処理を行う。唯一異なる点は、Balancer PC に対し、定期的に Keepalive パケットを送信することである。

Balancer PC の役割は、Join, Prune パケットを Branching PC に振り分けることである。Balancer PC は、Keepalive パケットによって稼働中の Branching PC を知り、担当 PC 表 (PC list) に記録する。担当 PC 表には、稼働中 Branching PC 一覧と、各

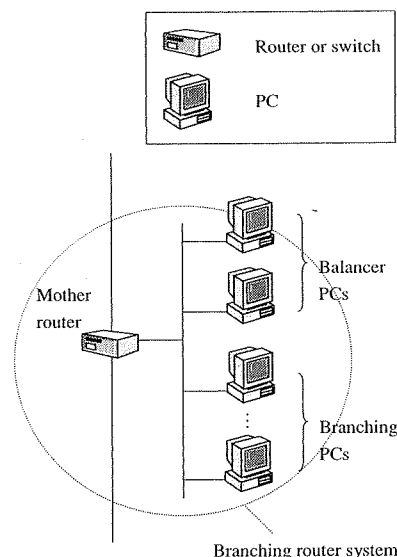


図 2 Branching ルータシステム構成例
Fig. 2 An example of branching router system.

Branching PC が担当しているチャンネルが記録される。一定時間 Keepalive パケットを受信しなかった Branching PC は、障害が発生したとみなされ、担当 PC 表から削除される。Balancer PC 間では、VRRP (Virtual Router Redundancy Protocol) [24] 等により、冗長化が行われる。

母ルータは、通過する UDP パケットのあて先ポート番号を検査し、Join, Prune パケットを Balancer PC へと転送する^(注6)。

4.3.2 負荷分散メカニズム

一般に、クラスタシステムでは、クラスタ入口に負荷分散装置を設置し、クラスタを構成する各コンピュータにパケットを振り分ける。トラフィック量に比べてデータ処理負荷が大きい場合には、このメカニズムは効果的に動作する。しかし、マルチキャストのようにトラフィック量が大きくなると、負荷分散装置が過負荷になるおそれがある。この課題を解決するために、サーバ直接応答 [6] という負荷分散方式を参考にする。詳細は省略するが、サーバ直接応答では、データパケットを負荷分散装置から迂回させる。Flexcast ルータシステムにおいても、Stream パケットが Balancer PC を通過しないよう留意して設計を行う。

本項では、Flexcast ルータシステムの負荷分散メカ

(注5)：なお、IP multicast は、ネットワーク層でマルチキャスト機能を実現するため、パケットのあて先を明示的に変更することが困難であり、クラスタシステムによるアプローチは適さない。

(注6)：近年では、多くのルータやスイッチが、このようなポート番号による転送機能をもつ。6.2.1 で述べるように、我々はいくつかの代表的なルータとスイッチで実験を行い、動作を確認している。

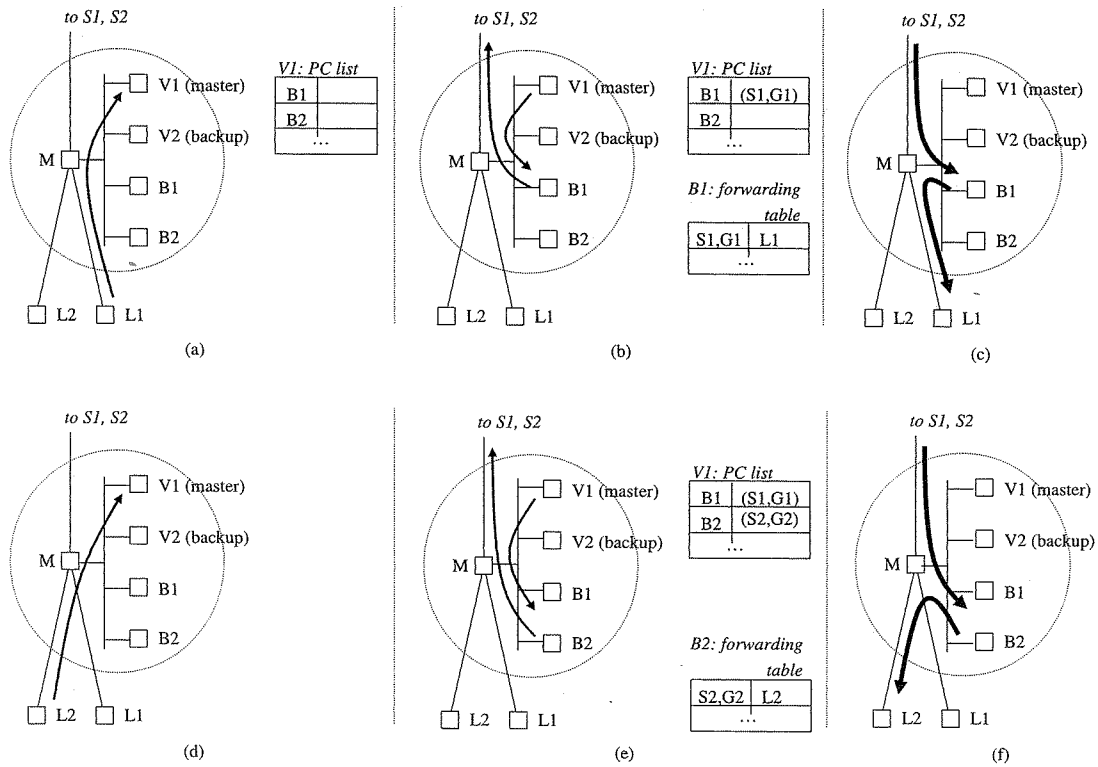


図3 Branching ルータシステム動作例

Fig. 3 An example of load balancing in branching router system.

ニズムを説明する. Balancer PC は, 母ルータから Join パケットを受信すると, 要求チャネルを担当している Branching PC を担当 PC 表から検索する. 担当が未割当である場合, 何らかのアルゴリズムによって, そのチャネルの担当 Branching PC を選択する. 選択アルゴリズムには, ラウンドロビンのような単純な方法から, クライアントの要求傾向に基づき負荷を先読みするような高度な方法まで, 任意のアルゴリズムを用いることができる. 最終的に, Balancer PC は Join パケットを担当 Branching PC へと転送する. このようにして, 各 Branching PC へと負荷を分散する.

Join パケットを受信した Branching PC は, Join パケットの送信元を転送表に登録する. そして, Branching PC は, 自分を送信元とする Join パケットを生成し, サーバへと送信する. すると, 上流からの Stream パケットは, Balancer PC を経由することなく, Branching PC へと直接送られてくる. このようにして, Stream パケットを Balancer PC から迂回させる.

ルータシステムとしての転送能力を向上するには, 単純に Branching PC を追加するだけでよい. 追加された Branching PC は, Balancer PC に Keepalive パケットを送信する. Balancer PC は, この Branching

PC を担当 PC 表に追加し, 次に担当を選択するときから候補として扱う. Branching PC を追加する際に, ルータシステムとしての機能を停止する必要はない.

図 3 に, Branching ルータシステムの動作例を示す. 図では, Leaf ルータ L1 がチャネル (S1, G1) に対し, Join パケットを送信している. 2 台の Balancer PC V1, V2 は, VRRP によって冗長化されている. VRRP については 4.3.4 で説明する. ここでは, V1, V2 のうち, V1 のみが動作していると考えればよい. V1 の担当 PC 表には, Branching PC B1, B2 が登録されている. 母ルータ M は, L1 が送信した Join パケットを V1 へと転送する (図 3(a)). Join パケットを受信した V1 は, 担当 PC 表を検索し, (S1, G1) が登録されていないことを知る. すると, 担当として B1 を選択し, Join パケットを転送する. B1 は, L1 を転送表に登録し^(注7), B1 を送信元とする Join パケットをサーバ S1 へと送信する (図 3(b)). Stream パケットが B1 に届くと, B1 は転送表に登録されている L1 へと転送する (図 3(c)).

続いて, Leaf ルータ L2 が, チャネル (S2, G2) へと Join パケットを送信する (図 3(d)). V1 は, 新た

(注7): 4.1 で説明したように, 実際には L1 の IP アドレスと送信元ポート番号が登録されるが, ここでは単に L1 とだけ記す.

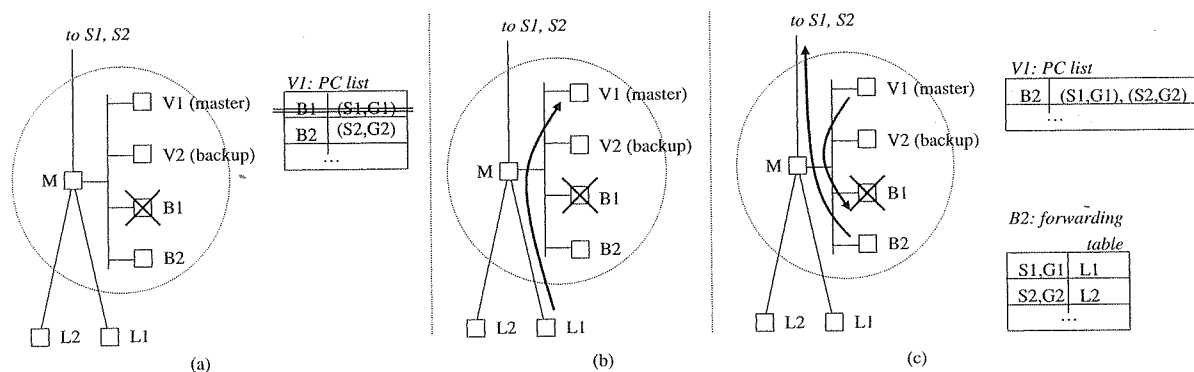


図 4 Branching PC 障害時の動作例

Fig. 4 An example of fail-over when a branching PC is getting down.

な担当として B2 を選択し、Join パケットを転送する。B2 は L2 を転送表に登録し、サーバ S2 へと Join パケットを送信する (図 3(e))。Stream パケットは B2 に到着し、L2 へと転送されていく (図 3(f))。このように、Balancer PC V1 によって、負荷は 2 台の Branching PC B1, B2 に分散される。

4.3.3 Branching PC 障害

本項では、Branching PC に障害が発生したときの動作メカニズムを説明する^(注8)。障害に見舞われた Branching PC の担当チャネルを、障害発生チャネルと呼ぶことにする。Branching PC に障害が発生すると、Keepalive パケットが途絶え、Balancer PC の担当 PC 表から削除される。その後、障害発生チャネルへの Join パケットを Balancer PC が受信すると、担当 PC 表の中から担当を選択し直す。そして、新担当によって、配信が再開される。

このように、Branching PC に障害が発生しても、ルータシステム全体としては Branching ルータの機能を継続することができる。障害発生チャネルの配信停止時間は、Keepalive がタイムアウトし、次の Join パケットを受信するまでとなる。停止時間を D とすると、次の不等式で表すことができる。

$$T_{keep} - I_{keep} < D < T_{keep} + I_{join} \quad (1)$$

ここで、 I_{keep} , T_{keep} はそれぞれ Keepalive パケット送信間隔とタイムアウト時間、 I_{join} は Join パケット送信間隔である。また、 $I_{keep} < T_{keep}$ である。

なお、Balancer PC にも Branching PC と同様の転送表を管理させることで、停止時間を短縮することもできる。Balancer PC は、Keepalive がタイムアウトすると同時に新担当を選択し、転送表を新担当に送信する。新担当は、すぐにサーバへと Join パケット送信し、配信を再開する。次の Join パケットを待たな

いため、停止時間を次の範囲に収めることができる。

$$T_{keep} - I_{keep} < D < T_{keep} \quad (2)$$

図 4 を用いて Branching PC に障害が発生した際の動作を説明する。Branching PC B1 に障害が発生すると、Keepalive パケットが途絶え、しばらくして Balancer PC V1 の担当 PC 表から削除される (図 4(a))。その後、チャネル (S1, G1) への Join パケットを受信した V1 は (図 4(b))、担当 PC 表から (S1, G1) を検索し、未登録であることを知る。新担当として Branching PC B2 を選択し、Join パケットを転送する。B2 は、転送表に (S1, G1) と L1 を登録し、サーバへと Join パケットを送信する (図 4(c))。

4.3.4 Balancer PC 障害

本項では、Balancer PC に障害が発生した場合を考える。Balancer PC の冗長化プロトコルには VRRP を用いるとする。VRRP では、複数の装置を一つのグループに所属させる。通常は、そのうち 1 台がグループを代表する仮想 IP アドレスを用いて通信を行う。通信を行っている装置をマスター、待機中の装置をバックアップと呼ぶ。マスター装置は定期的に Advertisement パケットを送信する。マスター装置に障害が発生し、Advertisement パケットが途絶えると、同一グループに属するバックアップ装置の一つが仮想 IP アドレスを引き継ぎ、マスターに切り換わる。母ルータや Branching PC は、仮想 IP アドレスに対して通信を行うため、切換後も ARP キャッシュの更新を行うだけで通信を継続することができる。

(注8)：本論文では、Keepalive パケットが途絶えるような障害を想定し、議論を行う。例えば、PC や OS、ネットワークケーブルに障害が発生した場合である。その他の障害、例えば、Branching ルータプロセスのみが停止するような障害については、既存のプロセス監視ツール等 [38] と組み合わせることで解決することができる。

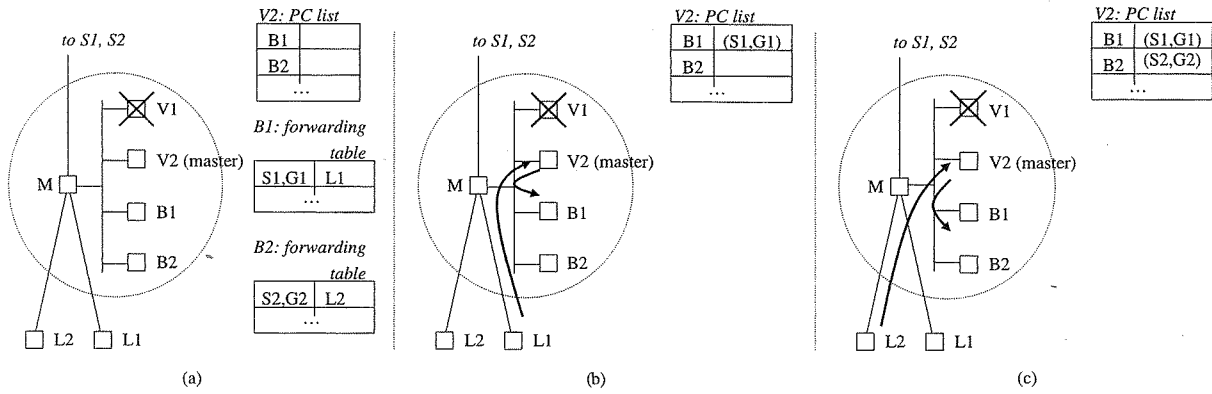


図 5 Balancer PC 障害時の動作例

Fig. 5 An example of fail-over when a balancer PC is getting down.

Balancer PC に障害が発生すると, Branching PC は一時的に Join パケットを受信できない状態になる. しかし, Join パケットがタイムアウトするまでに Balancer PC が切り換わり, 次の Join パケットが到着すれば, 配信が中断することはない. 最後の Join パケットを受信してから, 障害回復後に Join パケットを受信するまでの時間は, 次の時間の総和となる. 障害発生前最後の Join パケットを受信してから障害が発生するまでの時間, VRRP 切替時間, Keepalive を送信するまでの時間, 次の Join パケットが届くまでの時間. よって, 次の不等式が満たされるときには Join パケットがタイムアウトすることはない, ルータシステムとしては無瞬断で切替を行うことができる.

$$T_{vrrp} + I_{keep} + 2I_{join} \leq T_{join} \quad (3)$$

ここで, T_{vrrp} は VRRP 切替時間^(注9), T_{join} は Join タイムアウト時間である. Keepalive と同様に, $I_{join} < T_{join}$ である.

なお, 切替の前後で担当 Branching PC が変更される場合がある. このとき, 要求元でパケット重複が発生する. Balancer PC 間で担当 PC 表を同期しておくことで, この重複を避けることができる.

図 5 に Balancer PC V1 が障害に見舞われた例を示す. 障害発生後しばらくすると, VRRP によって Balancer PC V2 がマスターに切り換わる (図 5(a)). すると, 母ルータ M は V2 へと Join パケットを転送するようになる. V2 は, チャンネル (S1, G1) への Join パケットを受信し, 新担当を選択する. ここでは, Branching PC B1 を選択したとする. すると, Join パケットを B1 へと転送する (図 5(b)). 同様にして, チャンネル (S2, G2) への Join パケットを受信すると, 新担当として B2 を選択し, Join パケットを転送する

(図 5(c)). この間, 2 台の Branching PC B1, B2 は, 中断することなく配信を行うことができる.

4.3.5 Balancer/Branching PC

ここまで, Balancer PC と Branching PC を異なる PC として説明したが, 同一 PC に両方の機能を実装しても構わない. また, そのようにすることによって, 装置数を削減することができる. 本項では, 両機能をもつ PC によって構成された Branching ルータシステムについて説明する^(注10). 以降, Balancer PC と Branching PC の両機能をもつ PC を, Balancer/Branching PC と表記することにする.

負荷分散メカニズムは, 機能を分担した場合と同じである. VRRP によりマスターとなっている PC は, 各 PC から Keepalive を受信し, 担当 PC 表に記録する. このとき, 同一 PC から Keepalive を受信し, 担当 PC 表に記録する. 母ルータから Join パケットを受信すると, 担当 PC を選択し, 転送する.

続いて, 障害発生時の動作を説明する. Balancer/Branching PC に障害が発生すると, まず, VRRP によってマスター PC が切り換わる. 続いて, Keepalive によって担当 PC 表への登録が行われ, Join の振分けが開始される. このため, 障害発生チャネルの配信停止時間は, 次の不等式で表される.

$$T_{vrrp} < D < T_{vrrp} + I_{keep} + I_{join} \quad (4)$$

なお, 非障害発生チャネルへの配信は, 式 (3) が満たされている限り中断しない.

(注9): 厳密には, VRRP 切替時間は Advertisement 送信間隔と priority によって決定される [24]. しかし, 通常, 切替時間の幅は約 1 秒と小さいため, ここでは単純に T_{vrrp} と表す.

(注10): 通常, Join 処理負荷は Stream 複製処理に比べて圧倒的に小さいと予想されるため, Branching PC に Balancer 機能を追加しても, Stream 品質への影響は非常に小さいと考えられる.

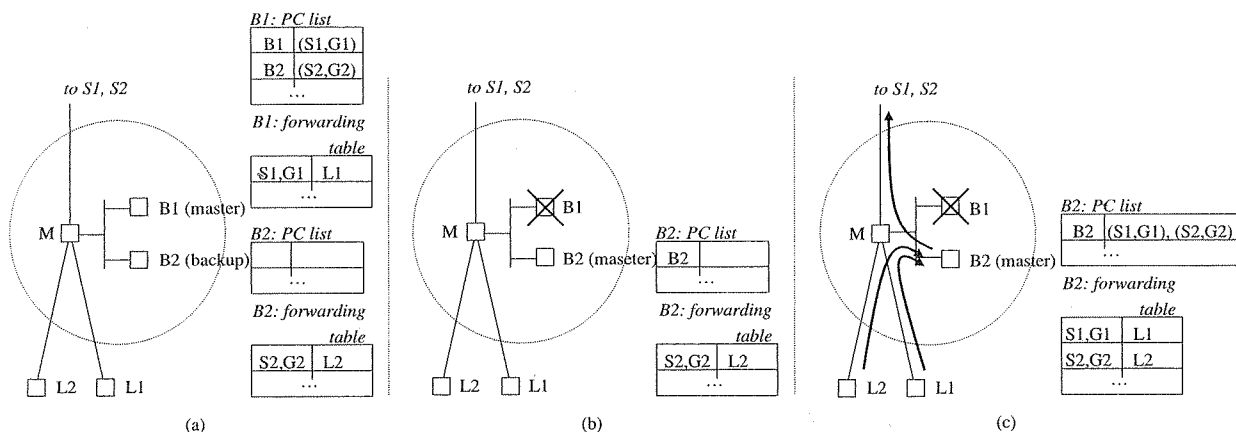


図 6 Balancer/Branching PC 障害時の動作例

Fig. 6 An example of fail-over when a balancer/branching PC is getting down.

図 6 の例を用いて、障害発生時の動作を説明する。図 6(a) は、障害が発生する前の様子である。Balancer/Branching PC B1 がマスターとなり、担当 PC 表を管理している。B1 がチャネル (S1, G1) を担当し、Balancer/Branching PC B2 がチャネル (S2, G2) を担当している。ここで、B1 に障害が発生したとする。すると、VRRP によって B2 がマスターになる。B2 は B2 自身に Keepalive パケットを送信し、担当 PC 表に登録する (図 6(b))。続いて、母ルータ M から (S1, G1) への Join パケットを受信すると、唯一の候補である B2 を担当として選択する。同様に、(S2, G2) への Join パケットも処理される (図 6(c))。このように、障害発生チャネルである (S1, G1) は、担当が変更された後に、配信が再開される。また、非障害発生チャネル (S2, G2) では、中断なく配信が続けられる。

4.3.6 Branching ルータ構成のまとめ

このように、クラスタシステムによって、スケーラビリティと可用性に優れた Branching ルータシステムを実現することができる。更に、母ルータを冗長化することで、このルータシステムは単一故障点をもたなくなる。

母ルータへの要求条件は、PC クラスタの接続とそれに伴う経路通知、Join、Prune パケットの転送設定である。なお、Branching PC は母ルータ以外の装置に接続することができる。例えば、母ルータの転送能力が低いときには、隣接する高性能スイッチに接続してもよい。

4.4 Root ルータシステム、Leaf ルータシステム

Root ルータシステムの構成は、Branching ルータシステムとほぼ同じである。図 7 に構成例を示す。図

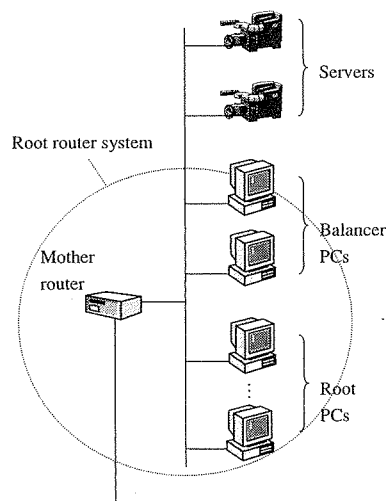


図 7 Root ルータシステム構成例

Fig. 7 An example of root router system.

のように、Root PC と同一セグメントにサーバを設置する。Balancer PC は Join を Root PC に振り分ける。Join パケットを受信した Root PC は、サーバが送出する IP multicast パケットを stream パケットにカプセル化し、下流へと転送する。その他の動作メカニズムは、Branching ルータシステムとほぼ同じである。

図 8 に Leaf ルータシステムの構成例を示す。Leaf ルータシステムの冗長化は、IGMP/MLD によって行われる。冗長化の基本原理は VRRP とほぼ同じであるため、ここでは省略する。負荷分散は、IGMP snooping [9] のようなデータリンク層のメカニズムによって行う。IGMP snooping では、IGMP Report の傍聴機能をもつスイッチを用いることで、IP multicast パケットの伝搬範囲を制限することができる。

なお、遠隔会議のように双方向でマルチキャスト

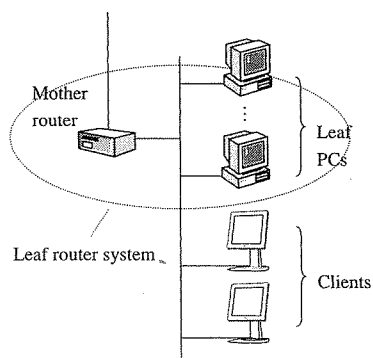


図 8 Leaf ルータシステム構成例

Fig. 8 An example of leaf router system.

を行うときには、Root ルータと Leaf ルータを同一母ルータに接続する。このとき、Root ルータと Leaf ルータの機能を同一 PC に実装してもよい。

5. グループ管理システムの設計

5.1 IGMP/MLD 後方互換性

2.2 で述べたように、Flexcast では、クライアント管理プロトコルに IGMPv3/MLDv2 を用いることを前提としている。これは、クライアントが送出する Report によって SSM チャンネルを特定するためである。しかし、現在はまだ IGMPv1,2/MLDv1 を実装したクライアントが多く存在する。IGMPv1,2/MLDv1 はグループアドレスのみを特定するプロトコルであり、Flexcast が用いている SSM チャンネルを特定するためには、サーバアドレスを追加するメカニズムが必要となる。本節では、グループアドレスをチャンネルへと変換する方式について議論を行う。

IP multicast では、この課題を解決するために二つの解決策を提案している。一つは、rendezvous point と呼ばれる仮想サーバを用いる方法である。この方法は PIM-SM [18] として標準化されている。もう一つは、URL rendezvous directory [37] と呼ばれる方法である。

PIM-SM は次のように動作する。Leaf ルータは Report を受信すると、rendezvous point へ Join パケットを送信する。そして、rendezvous point を根とする配信木を構築する。一方、Root ルータは、サーバから送出された IP multicast パケットを、ユニキャストパケットにカプセル化し、rendezvous point へと送信する。このようにして、サーバとクライアントは rendezvous point で出会い、rendezvous point を根とする配信木によって配信が開始される。その後、IP

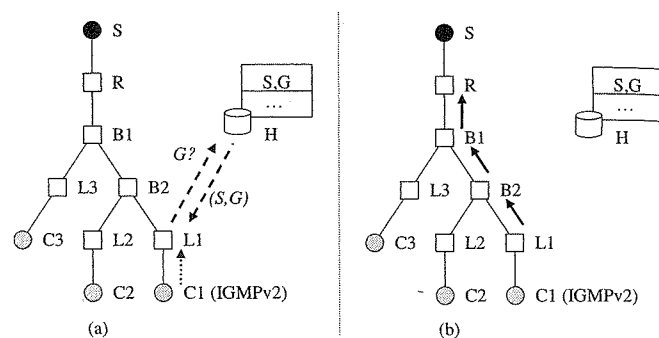


図 9 グループ管理システム動作例

Fig. 9 An example of group management.

multicast パケットが Leaf ルータに届くと、Leaf ルータは送信元アドレスからサーバアドレスを取得する。トラフィックがしきい値を超えると、Leaf ルータはサーバへ Join パケットを送信し、サーバを根とする配信木へと切り換える。このように、PIM-SM はメカニズムが複雑であり、また rendezvous point にトラフィックが集中するという課題もある [2]。

一方、URL rendezvous directory は、WWW サーバを設置し、そこにチャンネルを一元管理する方式である。クライアントは、Report を送信する前に WWW サーバに HTTP リクエストを送信する。このリクエストには、グループアドレスが含まれる。WWW サーバは、グループアドレスに対応するサーバアドレスを検索し、そのサーバアドレスを含む URL への再リクエストを促す。そして、クライアントは指示された URL へアクセスを試みる。ここで、Leaf ルータはこのアクセスを傍聴し、URL に含まれるサーバアドレスを記憶する。その後、クライアントが Report を送信すると、Leaf ルータは記憶したサーバへと Join パケットを送信する。URL rendezvous directory は、rendezvous point 方式に比べシンプルであるが、クライアントに URL rendezvous directory の実装を要求する。

本節では、それぞれの方式の課題を解決するシンプルなグループ管理方式を提案する。提案方式では、URL rendezvous directory と同様に、チャンネルを WWW サーバに一元管理する。この WWW サーバをグループ管理サーバ (group management server) と呼ぶ。Leaf ルータは、IGMPv1,2/MLDv1 Report を受信すると、グループ管理サーバにグループアドレスを送信し、対応するチャンネルを問い合わせる。

図 9 に示す例を用いて提案方式の動作を説明する。グループ管理サーバ H には、チャンネル (S, G) が登録されている。Leaf ルータ L1 には、H のアドレスが設

定されている。そして、L1 はクライアント C1 から Report を受信すると、グループアドレス G に対応するサーバアドレスを H に問い合わせる。H は L1 に S を返す。L1 は S に対して Join パケットを送出する。ここから先は、通常の Flexcast と同様に動作する。

グループ管理サーバに登録されているチャンネルには有効期限が設定されており、期限を過ぎたグループアドレスは再利用可能である。また、グループ管理サーバから取得したチャンネルにも有効期限が設定されているため、期限中の再問合せは不要である。提案方式には、rendezvous point のようなトラフィック集中点はなく、クライアントは IGMP/MLD のみを実装すればよい。なお、グループ管理サーバの冗長化については、一般の WWW サーバ可用性向上方式を利用することができる [6]。

最後に、表 2 に 3 方式をまとめる。比較項目は、追加装置 (additional node)、トラフィック集中 (traffic concentration)、クライアントへの追加要求 (requirements for clients) とした。

5.2 グループアドレス管理に関するスケーラビリティ

一般に、IP multicast では、グループアドレスをネットワーク全体で一意的に保つ必要がある。SSM チャンネルを用いることでこの課題を解決することが可能であるが、クライアント管理プロトコルに IGMPv3/MLDv2

を用いることが要求される。IGMPv1,2/MLDv1 からの過渡期においては、グループアドレスの一意性保証は、避けて通れない問題である。前節で提案したグループ管理方式と Flexcast を組み合わせることで、グループアドレスの一意性に関する条件を緩和することができる。本節では、この理由を説明する。

理解を容易にするために、次のようなシナリオを想定して説明を行う。マルチキャストネットワークを提供するあるネットワーク事業者と、その顧客を考える。顧客は、例えば、契約者に放送サービスを提供する二次プロバイダや、社内向けに遠隔会議や遠隔教育を行う企業である。ネットワーク事業者は Branching ルータを運用し、顧客は Root ルータ、Leaf ルータを運用する。

ここで、マルチキャストネットワークを PIM-SM によって運用する場合と、Flexcast を用いる場合とを比較する。Flexcast を用いる場合は、各顧客がグループ管理サーバを設置する。なお、顧客間でユニキャストアドレスの重複はないとする。

PIM-SM によってマルチキャストネットワークを運用する場合、顧客間でグループアドレスが重複しないように調整する必要がある。同一グループアドレスを使うと、異なる顧客のネットワークにパケットが混入する可能性がある。これは、PIM-SM ではグループアドレスのみで配信木を特定することがあるためである。

表 2 IGMP/MLD 後方互換方式比較
Table 2 Comparison of translation methods.

Translation method	Additional node	Traffic concentration	Requirements for clients
Rendezvous point	Rendezvous point	Yes	No
URL rendezvous directory	WWW server	No	URL rendezvous directory
Our proposal	Group management server	No	No

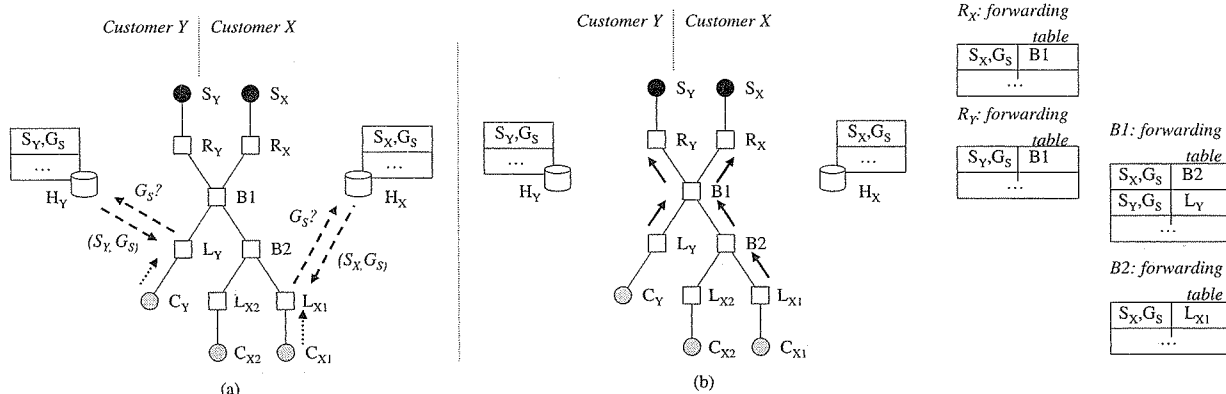


図 10 グループアドレス管理の独立性
Fig. 10 Independence of group management.

一方, Flexcast によってマルチキャストネットワークを運用している場合, グループアドレスは顧客内で一意であればよい. これは, Flexcast ルータは常にチャンネルによって配信木を特定するためである.

例えば, 図 10 のように, 顧客 X, Y が Flexcast によるマルチキャストサービスを受けているネットワークを考える. ネットワーク事業者は Branching ルータ B1, B2 を提供している. 顧客 X は, Root ルータ R_X , Leaf ルータ L_{X1}, L_{X2} , サーバ S_X , クライアント C_{X1}, C_{X2} , グループ管理サーバ H_X を運用している. 同様に, 顧客 Y は, Root ルータ R_Y , Leaf ルータ L_Y , サーバ S_Y , クライアント C_Y , グループ管理サーバ H_Y を運用している. ここで, 顧客 X, Y が共通するグループアドレス G_S を利用しているとする. C_{X1} と C_Y はグループアドレス G_S に対して Report を送信する. このとき, それぞれのグループ管理サーバは, G_S に対応するチャンネルとして, $(S_X, G_S), (S_Y, G_S)$ を返す. B1, B2 は別々のチャンネルとして認識し, 顧客間でパケットが混入することはない.

このように, 提案するグループ管理方式と Flexcast を組み合わせることにより, グループアドレス管理を顧客ごとに独立して行うことができる. ネットワーク全体で一意性を管理する場合に比べ, 管理が容易になり, スケーラビリティの向上が期待できる.

6. 実装と評価

4. で述べた設計方針に従って, Flexcast ルータを実装し, Branching ルータシステムの転送能力と障害発生時の切換時間を測定した. また, 5. で提案したグループ管理システムによって, 後方互換性が確保されていること, 顧客間で共通するグループアドレスを利用できることを確認した.

6.1 Flexcast ルータの実装

Flexcast ルータ機能と負荷分散機能を, Linux 上で動作するユーザ空間ソフトウェアとして実装した. 実装言語は C++ である. 負荷分散アルゴリズムはラウンドロビンとした. なお, Root ルータと Leaf ルータは, 同一ソフトウェアとして実装した. ソフトウェア開発環境を表 3 に示す.

また, 5. で提案したグループ管理サーバを, Java™ Servlet 技術 [42] を用いて実装した. データベースには, PostgreSQL [48] を用いた. 開発環境を表 4 に示す.

すべてのソフトウェアは, IPv4/6 両方の実装を行っ

表 3 Flexcast ルータ機能及び負荷分散機能開発環境

Table 3 Development environment of Flexcast routers and load balancing softwares.

Language	C++
Compiler	gcc 3.2.2
Operating system	RedHat Linux 9
Linux kernel	2.4.20

表 4 グループ管理サーバ開発環境

Table 4 Development environment of group management server.

Language	Java™ 1.4.1
Java™ Servlet server	tomcat 4.1.18
Database server	postgreSQL 7.3.1
Operating system	RedHat Linux 9
Linux kernel	2.4.20

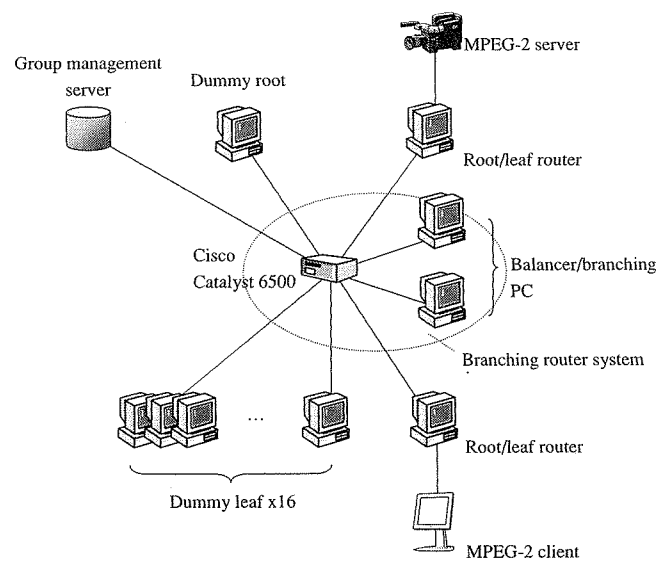


図 11 Branching ルータシステム評価実験ネットワーク

Fig. 11 Experimental networks for branching router system evaluation.

た. 本論文では, IPv4 で行った実験について述べる.

6.2 Branching ルータシステム性能評価

6.2.1 実験ネットワーク

図 11 に, Branching ルータシステム評価実験を行ったネットワークトポロジーを示す. サーバ, クライアントには, Kubotek 社製 MPEG-2 Transport Stream [45] を用いた. Branching ルータシステムへの負荷を増やすために, 擬似的な Root ルータ (dummy root) と擬似的な Leaf ルータ (dummy leaf) を作成した.

擬似 Root は, Root ルータと同様に転送表を管理する. また, 指定されたビットレートで IP multicast パケットを生成し, Stream パケットにカプセル化して送出することができる. 生成する IP multicast パ

ケットは 1,456 バイトとした。生成ビットレートについては後述する。擬似 Root は 1 台で複数のチャネルを送信することができ、多くのサーバが存在する状況を再現することができる。

擬似 Leaf は、指定されたチャネルに対し、Join パケットを送信する。また、10 秒間隔で Stream パケットの受信ビットレートを記録する。実験では、1 台の PC に複数の擬似 Leaf を起動し、多くのクライアントが存在する状況を模擬した。

図 11 に示すように、母ルータと 2 台の PC で Branching ルータシステムを構成した。2 台の PC には、どちらにも Balancer PC と Branching PC の両機能をインストールした。Keepalived for Linux [44] の VRRP 機能によって Balancer/Branching PC の冗長化を行った。母ルータには Cisco Catalyst 6500 を用いた^(注11)。指定されたポート番号に一致する UDP パケットを Balancer/Branching PC へと転送するように設定した。表 5、表 6 に、PC と母ルータの諸元を示す。

図 11 のように、Root/Leaf ルータはクラスタ化を行わなかった。MPEG-2 サーバ、クライアントを除き、すべて 1000Base-T のインタフェースを備える。MPEG-2 サーバ、クライアントは 100Base-TX である。実験パラメータを表 7 にまとめる。

表 5 Balancer/Branching PC 諸元
Table 5 Specification of balancer/branching PC.

CPU	Intel Xeon™ 3.06 GHz×1
Memory	512 MB DDR PC-21000 ECC Reg.×2
Mother board	Supermicro X5DPE-G2
Chipset	Intel E7501 chipset
PCI bus	PCI-X 64 bit/100 MHz
NIC	Intel PRO/1000MT Server Adapter

表 6 母ルータ諸元
Table 6 Specification of mother router.

Backplane	32 Gbit/s shared bus
Network Interface	1000Base-T×48
Operating System	IOS™ 12.2

表 7 実験パラメータ
Table 7 Parameters in the experiments.

Join interval I_{join}	10 s
Join timeout T_{join}	30 s
Keepalive interval I_{keep}	1 s
Keepalive timeout T_{keep}	4 s
VRRP fail-over time T_{vrrp}	4 s

6.2.2 転送能力評価

送受信ビットレートやチャネル数を変化させて実験を行い、Branching ルータシステムの転送能力を評価した。また、Balancer/Branching PC の数を変化させ、ルータシステムの拡張性についても評価を行った。

MPEG-2 サーバと擬似 Root の送信ビットレートは、6,184 kbit/s あるいは 20,000 kbit/s とした。それぞれ、SD、HD 規格の MPEG-2 Transport Stream を想定している。なお、MPEG-2 サーバ、クライアントは、送信ビットレートが 6,184 kbit/s のときのみ用いた。Balancer/Branching PC は、1 台あるいは 2 台とした。チャネル数は、2 あるいは 16 である。チャネルごとの要求数が等しくなるように擬似 Leaf を設定した。擬似 Leaf 数は結果グラフ中に示す。

次の手順で実験を行った。まず、Branching ルータシステムと Root/Leaf ルータ、MPEG-2 サーバ、擬似 Root を起動する。その後、MPEG-2 クライアントを起動し、続いて擬似 Leaf を起動する。すべての擬似 Leaf が受信を開始後、擬似 Leaf での受信ビットレートと Balancer/Branching PC の CPU 使用率を観測した。観測時間は 120 秒間とした。

結果を図 12～図 15 に示す。図 12 は、送信ビットレートを 6,184 kbit/s としたときに、擬似 Leaf で測定した受信ビットレート平均値である。横軸は、擬似 Leaf と MPEG-2 クライアントが要求するビットレートの総和である。以降、総要求ビットレートと呼ぶ。上側の横軸に、対応する分岐数を示す。図より、総要求ビットレートが 800 Mbit/s 以下の領域で

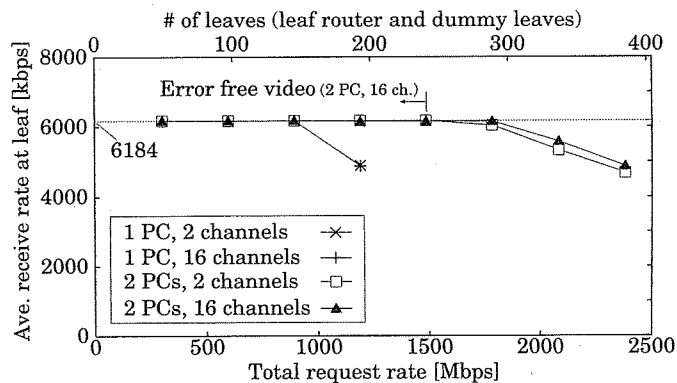


図 12 Branching ルータシステム転送能力評価 (送信ビットレート 6,184 kbit/s)

Fig.12 Evaluating forwarding ability of branching router system (data rate: 6,184 kbit/s).

(注11) : Cisco Catalyst 6500 以外にも、Cisco 7200, Foundry BigIron 4000 を用いて実験を行い、Branching ルータシステムを構成できることを確認している。

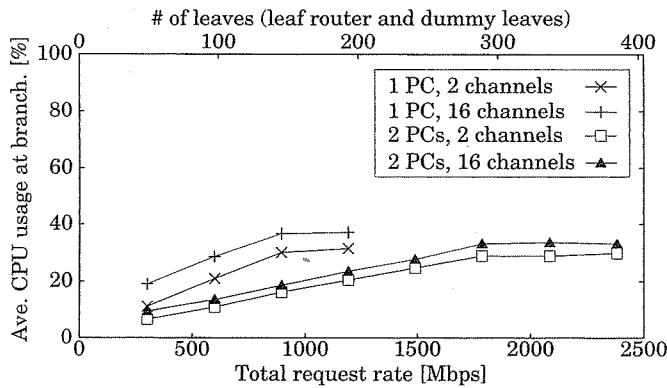


図 13 Balancer/Branching PC CPU 使用率 (送信ビットレート 6,184 kbit/s)

Fig. 13 CPU usage at balancer/branching PC (data rate: 6,184 kbit/s).

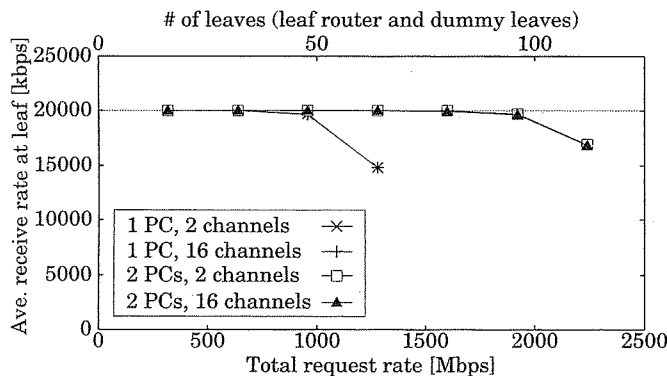


図 14 Branching ルータシステム転送能力評価 (送信ビットレート 20,000 kbit/s)

Fig. 14 Evaluating forwarding ability of branching router system (data rate: 20,000 kbit/s).

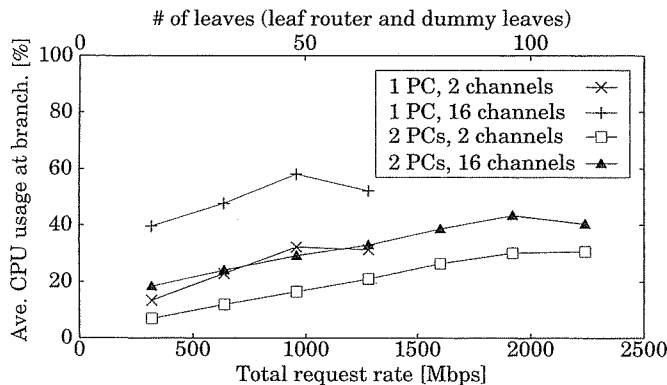


図 15 Balancer/Branching PC CPU 使用率 (送信ビットレート 20,000 kbit/s)

Fig. 15 CPU usage at balancer/branching PC (data rate: 20,000 kbit/s).

は、受信ビットレートが送信ビットレートに等しいことが分かる。このことは、Branching ルータシステムが、全要求にこたえられていることを意味している。また、Balancer/Branching PC を 1 台としたときに比べ、2 台にしたときは約 2 倍、1,500 Mbit/s の要

求にこたえられていることが分かる。図中の矢印は、Balancer/Branching PC を 2 台、チャンネル数を 16 としたときに、MPEG-2 クライアントがほとんど乱れなく映像を再生できた領域である。なお、チャンネル数の変化に対しては、有意な差は見られなかった。

図 13 は、送信ビットレートが 6,184 kbit/s のときに、Balancer/Branching PC で測定した CPU 使用率である。2 台の Balancer/Branching PC を用いたときには、VRRP によってマスタとなっている PC を測定した。横軸は図 12 と同じである。総要求ビットレートが増加するにつれて CPU 使用率は上昇するが、約 30~40% で飽和している。このときの総要求ビットレートは、図 12 において要求にこたえられなくなったビットレートと一致している。つまり、ボトルネックは Ethernet や PC 内部バス等、CPU 以外に存在していたと考えられる。なお、チャンネル数の増加に伴って、CPU 使用率が増加しているが、これは、入力トラヒックが増加するためであると考えられる。

図 14 は、送信ビットレートが 20,000 kbit/s のときの擬似 Leaf 受信ビットレートである。送信ビットレートを 6,184 kbit/s としたときと、同様の傾向を示している。

図 15 は、同じ送信ビットレートにおける Balancer/Branching PC の CPU 使用率である。送信ビットレートが 6,184 kbit/s のときと比べて CPU 使用率がやや高くなっているのは、入力トラヒックが増加したためと考えられる。

以上の結果より、PC クラスタによって構成した Branching ルータシステムは、Gbit/s レベルの高い転送能力を実現できることが分かった。更に、需要に合わせて PC を追加し、ルータシステムとしての転送能力を向上可能であることも明らかになった。

6.2.3 障害切換性能評価

本項では、Branching ルータシステムの障害切換性能について述べる。前項と同様、図 11 に示すネットワークを用い、実験を行った。

前項と同じ手順で各装置を起動する。障害切換を行うため、Balancer/Branching PC を 2 台とも起動しておく。その後、マスター PC のネットワークケーブルを抜き、Balancer/Branching PC の障害を模擬した。そして、擬似 Leaf で配信中断時間を観測し、障害切換性能を評価する。送信ビットレートを 6,184 kbit/s、チャンネル数を 16、総要求ビットレートを約 294 Mbit/s とした。擬似 Leaf 数は 48 であり、そのうち半数が障

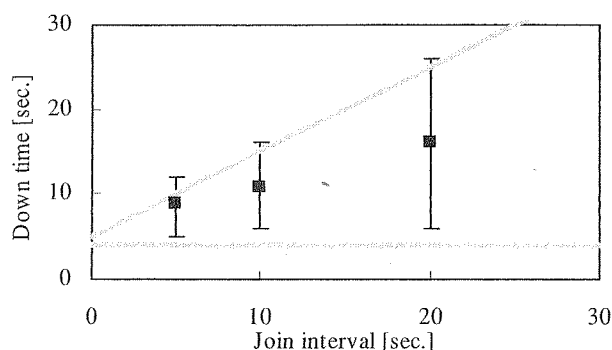


図 16 配信中断時間
Fig. 16 Down time.

害発生チャネルを受信する。

Join 送信間隔 I_{join} を 5, 10, 20 秒と変化させ、それぞれ 2 回ずつ測定を行った。Join タイムアウト時間 T_{join} は、Join 送信間隔 I_{join} の 3 倍とした。その他のパラメータは表 7 と同じである。

図 16 に結果を示す。横軸は Join 送信間隔 I_{join} 、縦軸は障害発生チャネルの配信中断時間である。グラフ中の点は中断時間の平均値を表し、誤差棒は最大値と最小値を表す。淡い太線は、式 (4) から予想される中断時間である。グラフより、おおむね予想時間内に配信が再開されていることが分かる。予想時間をわずかに超えているのは、ARP キャッシュ更新処理や Join パケット送受信処理、伝送遅延等の影響と考えられる。なお、本実験では式 (3) が満たされていたため、非障害発生チャネルが中断することはなかった。

以上の結果から、Branching ルータシステムは高い可用性をもつことが示された。

6.3 グループ管理システム評価

6.3.1 実験ネットワーク

図 17 に、グループ管理システムの評価実験を行ったネットワークトポロジーを示す。この実験ネットワークは、次のようなシナリオを想定して構築されている。あるネットワーク事業者が、Flexcast を用いてマルチキャストネットワークサービスを提供している。2 組の顧客 X, Y はネットワーク事業者と契約し、マルチキャストネットワークを用いて映像配信を行う。顧客は IGMPv2 クライアントを用いるため、グループ管理サーバを設置し、グループアドレスを SSM チャンネルへと変換する。ここで、顧客 X, Y は、偶然にも共通するグループアドレスを用いることとなったとする。

顧客 X は、サーバ、クライアントとして 6.2 で用いた MPEG-2 Transport Stream を用いた。顧客 Y は Windows Media [49] を利用した。なお、グループ

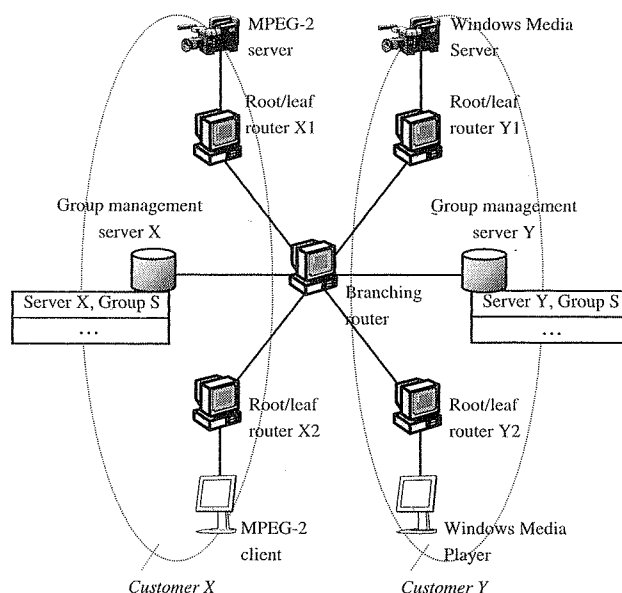


図 17 グループ管理システム評価実験ネットワーク
Fig. 17 Experimental networks for group management system evaluation.

#server	group	child
192.168.200.1	239.192.1.1	192.168.204.253:1025
192.168.205.10	239.192.1.1	192.168.203.253:1027

図 18 Branching ルータ転送表

Fig. 18 Forwarding table on the branching router.

管理システムの評価が目的であるため、Flexcast ルータのクラスタ化は行わなかった。

6.3.2 評価結果

前項で述べたシナリオに従い、グループ管理システムの評価実験を行った。それぞれのグループ管理サーバに、共通のグループアドレスをもつチャネルを登録しておく。そして、両顧客のクライアントが、それぞれ要求した映像を再生できることを確認する。

実験の結果、それぞれのクライアントでは、IP multicast パケットが混ざり合うこともなく、要求した映像を再生することができた。このとき、Branching ルータの転送表を見てみると、共通のグループアドレスをもつチャネルが別のエントリとして扱われていることが分かる (図 18)。なお、図中の child とは、要求元のことである。

このように、提案するグループ管理方式と Flexcast を組み合わせることにより、顧客間で共通するグループアドレスを用いることができる。グループアドレス管理にまつわるスケーラビリティ問題を大きく緩和できることが明らかになった。

6.4 Flexcast システムを用いたフィールド実験

本論文では、実験室内に構築した比較的小規模な

ネットワークでの実験結果を報告した。本節では、我々がこれまでに行った二つのフィールド実験について簡単に説明する。

一つは、Flexcast を用いた日米間遠隔会議実験である。この実験では、Internet2 [41] と GEMNet という二つの広域実験ネットワークを用い、地球規模の広域ネットワークでの配信が可能であることを確認した。

もう一つの実験では、研究開発用広帯域ネットワーク JGN (Japan Gigabit Network) [43] を用い、日本各地へのハイビジョン映像配信実験を行った。この実験は、MPLS ルータを用いた動的負荷分散方式 [30] の実験と同時にを行った。品質制御方式と組み合わせることにより、ハイビジョンのような広帯域ストリームであっても広域配信可能であることを実証した。

これらの実験では、Flexcast ルータのクラスタ化は行わず、6.3 の実験のように 1 台ずつの PC で Flexcast ルータを構成した。なお、実験の詳細は文献 [26], [46], [47] を参照されたい。

7. 関連技術

2.2 で述べたように、Flexcast は信頼性やセキュリティに関する機能をもたない。本章では、我々が開発したシステムと連携可能であると考えられる信頼性マルチキャスト方式やアクセス制御方式を簡単に紹介する。

7.1 信頼性マルチキャスト

Digital fountain アプローチ [7] は、誤り訂正符号によって信頼性を確保する方式である。ルータへの機能追加は不要であり、任意のマルチキャスト方式と組み合わせ利用することができる。我々は digital fountain アプローチを実装したソニー社製 Magnacast server を用いて、Flexcast との連携実験を行った。その結果、digital fountain アプローチによって、Flexcast に信頼性がもたらされることを確認している。また、6.4 で紹介したように、ルータによって実現される品質制御方式と連携することも可能である。

TCP トラヒックと共存するためには、ふくそう制御方式との連携も重要である。WEBRC (Wave and Equation Based Rate Control) は、end-to-end でふくそう制御を行う方式である [27]。連携実験は行っていないが、Flexcast との相性は良いと考えている。

7.2 セキュリティ

マルチキャストにおいては、暗号化によりデータ保護を行う方式 [21] が一般的である。通常、暗号化と複

合化はサーバとクライアントで行われる。このため、暗号化によるデータ保護は任意のマルチキャスト方式と組み合わせることができる。

一方、IGAP (Internet Group membership Authentication Protocol), MLDA (Multicast Listener Discovery Authentication protocol) は、IGMP/MLD Report に認証情報を追加し、Leaf ルータで認証、課金を行う方式である [22], [23]。我々は Leaf ルータに IGAP を実装し、アクセス制御実験を行った。実験の結果、Flexcast と連携可能であることを確認している。

8. むすび

本論文では、Flexcast を用いたマルチキャストシステムの開発について議論を行った。

まず、マルチキャストサービスの現状を分析し、初期投資が大きいという課題に着目した。そして、その原因の一つに IP multicast のアーキテクチャがあると考えた。IP multicast は、マルチキャスト機能をネットワーク層で実現している。しかし、このアーキテクチャは、対応ルータをネットワーク全体に展開することを要求する。そこで、マルチキャスト機能を上位層で実現している Flexcast を採用し、マルチキャストシステムを開発することとした。Flexcast は、対応ルータを段階的に展開することが可能なマルチキャスト方式である。更に、End-system multicast のようにトラヒック削減効果を犠牲にすることはなく、Xcast のように受信者数を制限することもない。

Flexcast ルータの開発にあたっては、WWW 検索システムや GRID computing で大きな成果を挙げているクラスタシステムに注目した。運用中のルータあるいはスイッチに小規模な PC クラスタを接続し、システムとして Flexcast ルータ機能を設計した。既存設備を有効利用し、開発コストや装置コストを抑えることで、初期投資を抑えることが可能となった。また、クラスタシステムにより、高いスケーラビリティと可用性を実現した。

Flexcast は、クライアント管理プロトコルに IGMPv3/MLDv2 を採用している。このため、IP multicast 用に開発されたクライアントを利用し、開発コストを抑制することができる。しかし、現在は IGMPv3/MLDv2 への移行過渡期であり、IGMPv1,2/MLDv1 を実装したクライアントもまだ多く存在する。本論文では、これまでに開発された

IGMPv1,2/MLDv1 クライアントを利用するために、グループアドレスから SSM チャンネルへの変換方式を提案した。更に、提案方式により、グループアドレス管理を顧客ごとに独立して行うことができることを示した。IP multicast のようにネットワーク全体で一意性を管理する場合に比べ、管理が容易になり、スケラビリティの向上が期待できる。

Flexcast ルータを実装し、転送能力と障害切換性能の評価実験を行った。その結果、Gbit/s レベルの転送能力を実現可能であること、高い可用性を備えることを明らかにした。更に、提案したグループ管理システムを実装し、後方互換性が確保されていること、顧客間で共通するグループアドレスを利用できることを確認した。

本システムによって、マルチキャストサービス開始時の経済的リスクを抑えることが可能となり、普及を促進できると期待している。

謝辞 Flexcast 研究開発プロジェクトを推進して頂き、論文執筆の機会を与えて下さった林一博グループリーダー（現 NTT アドバンステクノロジー）に深く感謝致します。

文 献

- [1] E. Aharoni and R. Cohen, "Restricted dynamic steiner trees for scalable multicast in datagram networks," Proc. INFOCOM'97, vol.2, pp.876-883, April 1997.
- [2] K.C. Almeroth, "The evolution of multicast: From the MBone to interdomain multicast to Internet2 deployment," IEEE Netw., vol.14, no.1, pp.10-20, Jan./Feb. 2000.
- [3] R. Boivie, N. Feldman, Y. Imai, W. Livens, D. Ooms, and O. Paridaens, "Explicit multicast (Xcast) basic specification," IETF Internet Draft, work in progress, June 2004.
- [4] A. Boudani and B. Cousin, "SEM: A new small group multicast routing protocol," Proc. ICT2003, vol.1, pp.450-455, Feb. 2003.
- [5] A. Boudani, A. Guitton, and B. Cousin, "GXcast: Generalized explicit multicast routing protocol," IETF Internet Draft, work in progress, March 2004.
- [6] T. Bourke, Server Load Balancing, O'Reilly & Associates, 2001.
- [7] J.W. Byers, M. Luby, M. Mitzenmacher, and A. Rege, "A digital fountain approach to reliable distribution of bulk data," Proc. SIGCOMM'98, pp.56-67, Aug. 1998.
- [8] B. Cain, S. Deering, B. Fenner, I. Kouvelas, and A. Thyagarajan, "Internet group management protocol, version 3," IETF Request for Comments 3376, Oct. 2002.
- [9] M.J. Christensen, K. Kimball, and F. Solensky, "Considerations for IGMP and MLD snooping switches," IETF Internet Draft, work in progress, May 2004.
- [10] Y.-H. Chu, S.G. Rao, and H. Zhang, "A case for end system multicast," Proc. SIGMETRICS'00, pp.1-12, June 2000.
- [11] J.C.-I. Chuang and M.A. Sirbu, "Pricing multicast communication: A cost-based approach," Telecommunication Systems, vol.17, no.3, pp.281-297, July 2001.
- [12] L.H.M.K. Costa, S. Fdida, and O.C.M.B. Duarte, "Hop by hop multicast routing protocol," Proc. SIGCOMM'01, pp.249-259, Aug. 2001.
- [13] S. Deering, "Multicast routing in internetworks and extended LANs," Proc. SIGCOMM'88, pp.55-64, Aug. 1988.
- [14] S. Deering, "An architecture for wide-area multicast routing," Proc. SIGCOMM'94, pp.126-135, Aug. 1994.
- [15] S.E. Deering, W.C. Fenner, and B. Haberman, "Multicast listener discovery (MLD) for IPv6," IETF Request for Comments 2710, Oct. 1999.
- [16] C. Diot, B.N. Levine, B. Lyles, H. Kassem, and D. Balensiefen, "Deployment issues for the IP multicast service and architecture," IEEE Netw., vol.14, pp.78-88, Jan. 2000.
- [17] H. Eriksson, "MBONE: The multicast backbone," Commun. ACM, vol.37, no.8, pp.54-60, Aug. 1994.
- [18] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C.-G. Liu, P. Sharma, and L. Wei, "Protocol independent multicast-sparse mode (PIM-SM): Protocol specification," IETF Request for Comments 2362, June 1998.
- [19] W.C. Fenner, "Internet group management protocol, version 2," IETF Request for Comments 2236, Nov. 1997.
- [20] R. Finlayson, "The UDP multicast tunneling protocol," IETF Internet Draft, Nov. 2003.
- [21] T. Hardjono and B. Weis, "The multicast group security architecture," IETF Request for Comments 3740, March 2004.
- [22] T. Hayashi, D. Andou, H. He, W. Tawbi, and T. Niki, "Internet group membership authentication protocol (IGAP)," IETF Internet Draft, Aug. 2003.
- [23] T. Hayashi, H. He, A. Tanabe, H. Satou, and T. Niki, "Multicast listener discovery authentication protocol (MLDA)," IETF Internet Draft, work in progress, May 2004.
- [24] R. Hinden, "Virtual router redundancy protocol (VRRP)," IETF Request for Comments 3768, April 2004.
- [25] H.W. Holbrook and D.R. Cheriton, "IP multicast

- channels: Express support for large-scale single-source applications," Proc. SIGCOMM'99, pp.65-78, Sept. 1999.
- [26] T. Inoue, S. Tani, K. Ishimaru, S. Minato, and T. Miyazaki, "Wide-area multicasting based on flexcast: Toward the ubiquitous network," Proc. APSITT'03, pp.301-306, Nov. 2003.
- [27] M. Luby, V.K. Goyal, S. Skaria, and G.B. Horn, "Wave and equation based rate control using multicast round trip time," Proc. SIGCOMM'02, pp.191-204, Aug. 2002.
- [28] G.S. Malkin, "RIP version 2," IETF Request for Comments 2453, Nov. 1998.
- [29] J. Moy, "OSPF version 2," IETF Request for Comments 2328, April 1998.
- [30] T. Ogura, J. Suzuki, A. Chugo, M. Katoh, and T. Aoyama, "Stability evaluation of a dynamic traffic engineering method in a large-scale network," IEICE Trans. Commun., vol.E86-B, no.2, pp.518-525, Feb. 2003.
- [31] S. Ohta, S. Tani, and T. Miyazaki, "Multicast as a traffic variance smoother for IP streaming service," Proc. Networks'02, pp.105-110, June 2002.
- [32] G. Phillips, S. Shenker, and H. Tangmunarunkit, "Scaling of multicast trees: Comments on the chuang-sirbu scaling law," Proc. SIGCOMM'99, pp.41-51, Aug. 1999.
- [33] I. Stoica, T.S.E. Ng, and H. Zhang, "REUNITE: A recursive unicast approach to multicast," Proc. INFOCOM'00, vol.3, pp.1644-1653, March 2000.
- [34] S. Tani, T. Miyazaki, and N. Takahashi, "Adaptive stream multicast based on IP unicast and dynamic commercial attachment mechanism: An active network implementation," Proc. IWAN2001, pp.116-133, Sept. 2001.
- [35] D. Thaler, M. Talwar, A. Aggarwal, L. Vicisano, and D. Ooms, "IPv4 automatic multicast without explicit tunnels (AMT)," IETF Internet Draft, work in progress, Feb. 2004.
- [36] R. Vida, L.H.M.K. Costa, S. Fdida, S. Deering, B. Fenner, I. Kouvelas, and B. Haberman, "Multicast listener discovery version 2 (MLDv2) for IPv6," IETF Request for Comments 3810, June 2004.
- [37] Cisco systems, Inc., <http://www.cisco.com/>
- [38] daemontools, <http://cr.yp.to/daemontools.html>
- [39] Google, <http://www.google.com/>
- [40] Global Grid Forum, <http://www.ggf.org/>
- [41] Internet2, <http://www.internet2.edu/>
- [42] Java Servlet Technology, <http://java.sun.com/products/servlet/>
- [43] Japan Gigabit Network, http://www.jgn.nict.go.jp/english/index_E.html
- [44] Keepalived for Linux, <http://keepalived.sourceforge.net/>
- [45] Kubotek Corp., <http://www.kubotek.com/enginfo/Ehome.html>
- [46] NTT News Release, "NTT streams super high definition movies in the global-scale high-speed network," <http://www.ntt.co.jp/news/news02e/0211/021113.html>, Nov. 2002.
- [47] NTT News Release, "The successful demonstration of simultaneous wide-area multipoint transmission of HD streams for the broadband ubiquitous era," <http://www.ntt.co.jp/news/news03e/0305/030530.html>, May 2003.
- [48] PostgreSQL, <http://www.postgresql.org/>
- [49] Windows Media, <http://windowsmedia.microsoft.com/>
- (平成 16 年 5 月 18 日受付, 9 月 13 日再受付)

井上 武 (正員)



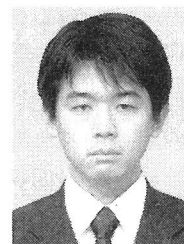
1998 京大・工・物理工学卒。2000 同大大学院工学研究科修士課程了。同年 NTT 入社。以来、移動通信、マルチキャスト通信の研究に従事。現在、未来ねっと研究所社員。2002 本会情報ネットワーク研究会研究賞受賞。

谷 誠一郎 (正員)



1993 京大・工・情報工学卒。1995 東大・理・情報科学専攻了。1995 日本電信電話株式会社入社。現在、同社コミュニケーション科学基礎研究所研究主任。通信プロトコル、離散アルゴリズム、量子計算・通信の研究に従事。2000 年度情報・システムソサイエティ論文賞(先見論文)受賞。情報処理学会, ACM, IEEE 各会員。

高橋 宏和 (正員)



2000 長岡技科大・工・電気電子システム工学卒。2002 同大大学院・工学研究科修士課程了。同年 NTT 入社。以来、マルチキャスト通信の研究に従事。現在、未来ねっと研究所社員。



湊 真一 (正員)

1988 京大・工・情報卒, 1990 同大大学院修士課程了. 1995 同博士後期課程(社会人選抜)修了. 博士(工学). 1990 NTT 入社. 以来, 大規模論理データの表現と処理アルゴリズムの研究開発に従事. 1997 米国スタンフォード大客員研究員. 1999 NTT 未来ねっと研究所主任研究員. 2004 より北大情報科学研究科助教授. 著書 “Binary Decision Diagrams and Applications for VLSI CAD” (Kluwer, 1995). 2000 情報処理学会山下記念研究賞受賞. IEEE, 情報処理学会各会員. 2000~2003 情報処理学会論文誌編集委員.



宮崎 敏明 (正員)

1981 電通大・電気通信・応用電子卒. 1983 同大大学院修士課程了. 同年日本電信電話公社(現 NTT)入社. 現在, 同社未来ねっと研究所グループリーダー. LSI-CAD, 通信用 FPGA 及び応用装置, アクティブネットワーク, ユビキタスネットワークの研究開発に従事. 新潟大大学院客員教授. 東京農工大非常勤講師. 工博(東工大). IEEE, 情報処理学会各会員.



豊島 鑑 (正員)

1982 福井大・工・電子卒. 1984 同大大学院工学研究科修士課程了. 同年日本電信電話公社(現 NTT)入社. 以来, ATM 伝送・交換システム, IP over ATM システム, Network ベース DDoS 防御システム, Unicast ベース多地点配信システム等の研究開発に従事. 現在, NTT 未来ねっと研究所主任研究員. 1988 本会篠原記念学術奨励賞受賞. IEEE 会員.