

第IX部

Explicit Multicast

第9部 Explicit Multicast

第1章 はじめに

明示的マルチキャスト (XCAST: Explicit Multicast) は、従来の ISM (Internet Standard Multicast) でのグループアドレスに変えて、パケットヘッダに到達すべき複数のマルチキャストアドレスを明記することで宛先を指定するマルチキャスト方式である。XCAST は ISM にくらべてグループメンバー数に制約がある一方で、グループ数に関するスケーラビリティに優れており、多地点ビデオ会議やネットワーク対戦型ゲームなど、多数のエンドユーザがプライベートなマルチキャストグループに対して発信が必要な用途に有効である。

WIDE project では 1999 年度に v6 WG の活動として XCAST の研究開発を開始し、独自に提案した Internet Draft MDO6 (Multiple Destination Option on IPv6) を元の実装 MDO6-kit の作成と試用を WIDE 6Bone 上で行って来た。並行して 2000 年末に IBM、Alcatel 等と共同で統一 XCAST プロトコルを策定し Internet Draft として提案した。その後、2000 年 11 月より WIDE 内で WG として独立した。

本報告書では 2001 年度の XCAST WG の活動を以下の順に報告する。

XCAST6 の実装

XCAST 統一提案を元にした IPv6 向け XCAST の実装

試用実験

MDO6/XCAST6 両実装を用いたプロトコル有用性検証のための試用実験。

メンバー管理機構の作成

XCAST のための、集中サーバーを排した分散型メンバー管理機構作成。

Reliable XCAST

XCAST を用いた信頼性のあるトランスポート層のプロトコルの提案

第2章 XCAST の実装

2.1 XCAST6 プロトコル

XCAST は、Alcatel、IBM、WIDE Project/富士通研究所が共同で提案している小規模グループのための明示的マルチキャストプロトコルで、2000 年に Internet Draft (draft-ooms-xcast-basic-spec-xx.txt、XCAST basic specification) として共同で寄書を行った。XCAST は 1999 年に上記の 3 社が独立に提出した提案 (CLM: Connectionless Multicast、SGM: Small Group Multicast) を統合したもので、以下のような特徴を持つ。

- マルチキャスト宛先をユニキャストのリストとして持つ
- IPv4、IPv6 の両方のプロトコルをサポート
- 配送状況の bitmap による表現
- ポートリスト
- DSCP リスト
- 経路キャッシュ-ISM との整合性を考慮した channel ID
- core network での対応を待たずに end-to-end の使用開始を可能にする漸近的導入のための機構
- X2U (xcast to unicast) 変換による XCAST 非対応ノードでのパケット受信

本 Internet Draft を元に、IBM、Alcatel、WIDE Project/富士通に ETRI/KAIST を加えた 4 つの団体が実装と相互接続時実証に向けて作業を行っている。WIDE Project は 2001 年度に XCAST の IPv6 部 (以下 XCAST6) の実装を行い、6Bone 上での試用実験を行った。

2.2 XCAST6 実装

XCAST 実装は以下の 3 パートからなる。

- kernel patch & library
- application patch
- xcgroup

WIDE project における XCAST6 の実装は Net BSD 1.5.2 への改造として作成された。NetBSD の IPv6 実装は WIDE Project の KAME の実装が採り入れられている。作成規模は kernel stack 2215 line、library 2341 line である。改造規模をこの程度に押えることができたのは、XCAST が経路管理機構を必要とせず、ユニキャストの経路管理機構に完全に依存し、中継処理中の経路検索処理などを KAME が既に用意していた関数群を完全に流用して実現したからである。

プロトコル実装を用いて以下の MBone アプリケーションを XCAST6 に対応させた。

- vic: 多地点画像配信アプリケーション
- rat: 多地点音声配信アプリケーション

アプリケーションの XCAST 対応は非常に単純で、基本的に以下の2つの作業を行った。

ユーザが明示的に宛先アドレスリストを指定できるインターフェースの追加 (ex. アプリケーションの起動引数)。

パケット送出処理のための一連の socket API のうち、ISM に対する connect()、sendmsg() を対応する xcast API (XcastCreateGroup(), XcastSendMsg()) などへ置き換え、上記の宛先リストを各 API に与える。受信処理や RTP などのトランスポート層についてはなんら変更を加える必要はない。

一方で、上述した単純な XCAST 対応では、ユーザがアプリケーション起動時に宛先リストを入力したり、メンバに変更があったときに再起動するなどの不便を強いることになる。この負担を軽減するため簡易メンバーシップ管理機構 xcgroup を作成した。xcgroup は、ruby script で書かれたサーバプログラムとクライアントプログラムとから構成される。サーバプログラムは、web CGI として動作し、グループ名とクライアントの IP アドレスの対応を管理する。クライアントプログラムは、グループ名と自身の IPv6 アドレスを登録する CGI リクエストを送出し、返り値としてグループに登録されている最新のクライアントの IPv6 アドレスリストを得る。この処理をすべてのクライアントが定期的におこなうことで、全体がクライアントリストを共有する。これにより、URL をグループ ID とした自動的なメンバー管理が可能になった。ただし、メンバーの参加までのタイムラグが CGI 要求の間隔に依存したり、コミュニティ自身を創出する仕組みについて考慮が

ないなどの欠点がある。

作成した実装は以下の URL より BSD like なライセンスの元で広く公開されている。

<http://www.sourceforge.net/projects/xcast6>

第3章 試用実験

XCAST の有用性を実証するために、2000 年度より 6Bone 上での試用実験を行っている。2001 年度の試用実験では以下のトピックがあった。

- 太平洋回線をまたいだミーティングの実施
- XCAST6 へのプロトコル移行
- ADSL などのより手軽なアクセスラインを用いたユーザーの収容

3.1 太平洋回線をまたいだミーティングの実施

2000 年度からの週1回の定期ビデオミーティングを継続して行った。2001 年度は、XCAST WG のメンバーが米国 University of California Irvine へ留学派遣され、現地より参加した。

UCI ネットは当初 IPv6-over-IPv4 tunneling で慶應義塾大学湘南藤沢キャンパスのルータに収容された。SFC 側のルータは MDO6 対応でなかったため、日本の WIDE 6Bone 上に2つ以上の参加者がいる場合、宛先リストの構成によっては太平洋回線を MDO6 パケットが往復し、通信品質の劣化がたびたび発生した (図 3.1 左)。

そこで、WIDE 6Bone より NLA2 prefix の割り当て(2001:200:161:8000::/48)を受け、SFC に XCAST WG で tunneling 用の MDO6 をインストールした gateway を設置し、ここに UCI を収容した。これにより、国内からの MDO6 トラフィックは必ず SFC で分岐し、太平洋線の無用なパケット往復は解消された (図 3.1 右)。

ミーティングでは主に、XCAST WG の自身の運営議論、修士研究の相談を行った。また、2001 年 6 月 7 日に幕張メッセで開催された BSD なひととき (NETWORLD + INTEROP 2001 TOKYO 併設の BSD ユーザ会) において Open XCAST Meeting と題して、UCI より XCAST 技術の紹介を遠隔プレゼンテーションとして行った。

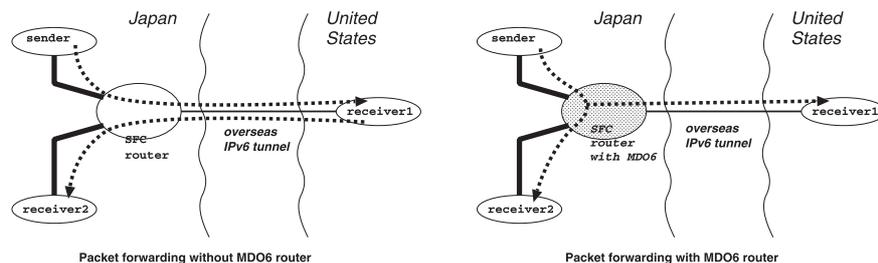


図 3.1. MDO6 ルータ導入によるパケット往復の解消

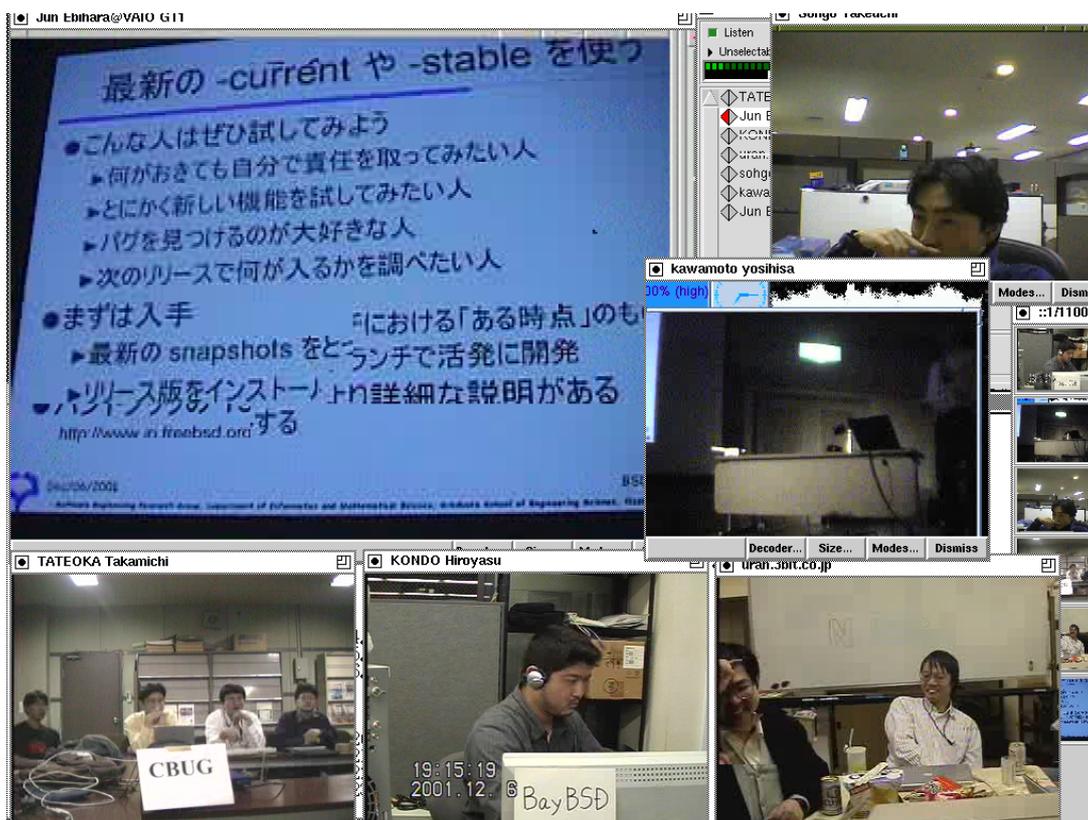


図 3.2. BSD なひととき中継の様子

また 2001 年 12 月 6 日にパシフィコ横浜で開催された BSD なひととき (Internet Week 2001 併設) においても XCAST 中継を行った。WIDE 6Bone 外より Bay BUG、NoBUG、CBUG 等の BSD ユーザグループを中継に収容し、Inter-domain multicast が手軽に行える事を実証した。

3.2 ADSL などより手軽なアクセスラインを用いたユーザーの収容

10/22 より WG メンバーを、SFC に設置した XCAST 対応 tunneling ルータで自宅 ADSL ラインよりミーティングに収容する実験を開始した。多地点会議では、送出する画像音声データが 1 セッ

ションに対して、受信データは複数である。これは、ADSL 等における代表的な帯域割り付けである、受信帯域が広い割り付けに合致する。しかし、通信経路上の適切な位置に XCAST 対応ルータが存在しない場合には、エンドノードで受信データを次の受信者に転送する必要が生じ、通信量がこの割り付けに合致しなくなる。ADSL 前の最終段に XCAST ルータを設置することで、非対称アクセスラインの帯域割り付けに合致させることができ、帯域の有効利用が可能になった。

各ユーザに対しては Dynamic Tunnel Configuration Protocol を用い、/60 の network prefix を静的に配付している。また、WIDE XCAST WG

へ割り当てられた NLA2 アドレスブロックのうち 2001:200:161:8000::/49 を tunnel 収容用に用いている。

3.3 XCAST6 へのプロトコル移行

XCAST6 統一提案を元にした実装の完成に伴い、2001 年 11 月 13 日より、試用実験システムも XCAST6 ベースに移行した。移行にともない xcgroup による簡易メンバーシップ管理サーバを SFC NOC 内に設置し使用を開始した。ミーティング直前の宛先 IPv6 アドレス交換の手間がなくなり、使い勝手が飛躍的に向上した。

第4章 メンバー管理機構の提案と実装

XCAST6 はネットワーク層ではグループの管理もメンバーの管理も行わないため、アプリケーション層が明示的にメンバーを管理する必要がある。XCAST WG では、昨年度アウェアネスをベースにしたメンバーグループ管理機構を作成し、今年度は XCAST6 実装に簡易メンバーシップ管理システム xcgroup を付加して提供している。両者は、インターネット上にメンバー管理サーバを設置し、XCAST6 アプリケーションが個別にメンバー情報の登録・取得を行う集中型の構成を採っている。集中型は、構成が平易で理解や作成がしやすい反面、特定のノードに管理を依存するため、アドホック性や対故障性に欠ける。

そこで、上記の点を解消することを目的に、分散型グループ管理機構を作成した。分散型管理機構を用いることで、グループの管理を特定のノードに依存せず、XCAST6 に参加するノードだけでメンバー管理を完結させることが可能になった。

メンバー管理システムはグループに参加するノードが、制御チャンネルをリンクとするメンバー管理のためのネットワークを構成し、隣接する制御メッセージを制御チャンネルを通じてお互いに交換し合うことで、全体としてメンバーのリストを得る。グループに参加したいノードのユーザはアプリケーション起動時に隣接ノードのアドレスと、メンバー管理情報を受け付けるポートの組を指定する。参加ノードが、

このポートへ参加要求を行うことで、グループ内への他のメンバーへの IPv6 アドレスの紹介が、メッセージのチャンネル中継を経て起こる。

各ノードの到達性保証は定期的な keep alive メッセージで行う。ノードの脱退や故障による制御チャンネルネットワークグラフの分離は自動的に検出され、グラフが再構成される。これにより集約型メンバー管理で起こるサーバ故障による XCAST 通信の停止を防止している。

第5章 Reliable XCAST

インターネットの利用者数が増加し続けるにともなって、WWW サービスやファイル配信サービス等のサーバは、非常に多くの要求を処理しなければならないようになった。しかし、1 台のサーバで同時に処理できる数には限りがある。また、利用者側から考えた場合、サーバが 1 台しかなければ、サーバの状態やサーバまでの距離、ネットワークの状態に関わらず、そのサーバに要求を出すしかない。そこで最近では CDN (Contents Distribution Network) が注目されている。CDN とはコンテンツを利用者に効果的に配信するために分散して配置されたミラーサーバとその制御システムの集合体である。

複数用意するミラーサーバのあいだでミラーリングを行う際の通信方法は 1 対 1 通信 (ユニキャスト) を複数回行うよりも、1 対多通信 (マルチキャスト) を 1 回行う方が帯域を有効に使用できる。しかしこれまでの IP マルチキャストでこれを行うには、グループをマスターサーバからのパケットを受信する複数のミラーサーバの組み合わせごとに 1 つずつ形成しなければならず、従来のグループマルチキャストがグループ数に対して規模適応性がないことが致命的欠陥となる。XCAST はグループ数に対する規模適応性が非常に優れている。これを活用し CDN のミラーリングを念頭におき、XCAST を用いたミラーリングのためのファイル配信プロトコルを設計し、挙動をシミュレーションし有用性を検証した。

5.1 プロトコル概要

マルチキャストを用いた高信頼性トランスポート

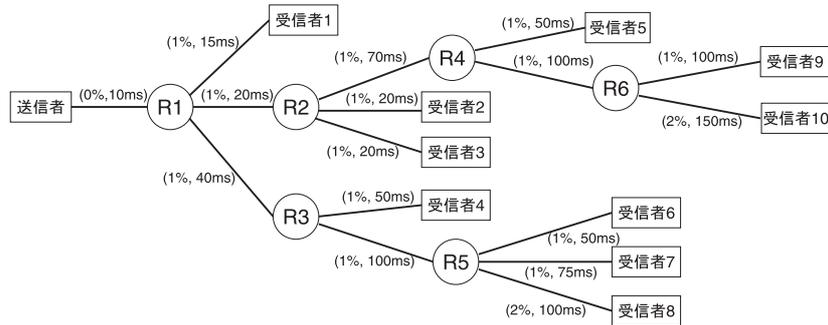


図 5.1. 実験用ネットワーク

プロトコルは、ネットワーク層以下でのパケットの消失を補うための仕組みとして、FEC (Forward Error Correction) と ARQ (Automatic Repeat reQuest) のどちらか、およびその組み合わせを用いる。FEC は送信する際に冗長なデータを送信し、受信者はパケットの紛失を検出すると、冗長なデータから紛失したパケットを再生する。これに対して ARQ は受信者がパケットの紛失を検出し、該当パケットの再送を受信者に要求する方法である。

今回の研究では、ファイル配信を目的のひとつとしているため ARQ を用いている。FEC では複数のパケットが紛失した際に必ずしも元のデータが再現できるとは限らないのに対して、ARQ は受信者がデータが完全になるまで再送を要求することで、完全な復旧が可能であるためである。

ARQ を採用した場合、再送要求と再送をどのように行うかが、高信頼性マルチキャストごとの特徴に大きく影響を与える。再送要求は、単純な ACK (Acknowledgement) により行う場合、ACK に NACK (Negative Acknowledgement) を併用する場合、SACK (Selective Acknowledgement) を使用する場合が代表的である。本研究ではこの 3 者をシミュレーションで評価し、最終的に SACK を採用した。

また再送も、マルチキャストで行う方法とユニキャストで行うのが代表的であり、今回はさらに XCAST で行う選択肢が増えている。XCAST 配送木の根元付近でパケットの消失が起こった場合、消失点から下流の複数のノードへ再送が必要な場合がある。この場合、取りこぼしが報告されたノードを明示的に指定してできることから、XCAST で再送することで、配送を効率化できると考え今回はこの方法をプロトコルに採用した。

これにより、プロトコル全体の大きな流れは以下

のようになった。

1. 送信者は XCAST で全受信者にパケットを送信する。ミラーリングや分散協調作業といった応用を考えた場合、グループ数に対する規模適応性が必要である。そのため、XCAST でパケットを送信する。
2. 受信者はパケットの受信/取りこぼし状態を SACK (Selective Ack) を用いて送信者に通知する。SACK を用いることで、受け取ったパケットと受け取っていないパケットの両方の情報を載せることができる。
3. 送信者は SACK を受け取って紛失があるとわかってもすぐに再送をせず、しばらく他の受信者からの SACK を収集した後、XCAST で再送する。

5.2 シミュレーションによる評価

本プロトコルの性能を評価するため、シミュレーションにより 1 回目の送信形態、再送アルゴリズムを変化させ、プロトコル間の比較を行った。本稿では、本送信にも再送にもユニキャストのみを用いた場合と XCAST を用いた場合の比較を示し、XCAST によるトラフィック量削減効果を見る。

シミュレーションで仮定したネットワーク環境は以下のとおりである (図 5.1)。

丸や四角で囲んでいるのが送信者や受信者、ルータ等のノードであり、それらを結ぶ線がリンクである。丸で囲まれた R1 から R6 はそれぞれ XCAST 対応ルータを表す。各リンクに付いている括弧の中の 2 つの値は、それぞれ紛失率と遅延を表す。

図では送信者と R1、R1 と受信者 1 などは隣接しているが、これはパケットが分岐しないルータを省略してある。送信者と R1 の間にルータを入れたと

表 5.1. ユニキャストとの比較

プロトコル	最速 受信者 終了時間 [s]	最遅 受信者 終了時間 [s]	ユニキャスト 再送の場合の 理想再送回数	送信者の 実際の 再送回数	受信者の 送信回数	本送信以外の パケット数
ユニキャスト	8.40	11.42	2846	2947	247	3194
本プロトコル	5.65	7.96	2849	1688	69	1757

表 5.2. ユニキャストとのトラフィック量の比較

プロトコル	本パケット (再送パケット) の大きさ [bytes]	再送要求 パケット の大きさ [bytes]	送信者の 全送信パケット数	受信者の 送信パケット数	全体の トラフィック量 [bytes]
ユニキャスト	560	560	102,947	247	57,788,640
本プロトコル	776	560	11,688	69	9,108,528

しても、このルータがパケット分岐点でなければ本プロトコルの振舞は変わらない。

このネットワークにおいて 512 Bytes の連続したデータフラグメントを 10000 個送信する場合のパケット数とトラフィック量についてユニキャストのみで行った場合と、XCAST を使用した本プロトコルとを比較した結果を以下に示す。なお、再送要求にはともに SACK を使用している。

この比較より XCAST を用いることの利点、欠点を以下のように読み取ることができ、ユニキャストよりも XCAST を用いたほうが高信頼性を効率的に実現できたと考えられる。

- 利点
 - － 終了時間が早い。
 - － 送信者の送信回数が非常に少ない。
 - － 本送信以外のパケット数が少ない。
 - － 全体のトラフィック量が非常に少ない。
- 欠点
 - － 送信者が送信するパケットの大きさが大きい。