

第 XVI 部

Explicit Multi-Unicast

第 16 部

Explicit Multi-Unicast

第 1 章 はじめに

Explicit Multi-Unicast 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ユニキャストアドレスを明記することで宛先を指定する配送方式である。XCAST は ISM に比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ー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する事が必要な用途に有効である。

WIDE プロジェクトでは 1999 年度に XCAST の研究開発を開始して以来、IBM、KAIST、Soongsil University、IRISA などとの協調により、統一 XCAST プロトコルの策定と Internet Draft の提案、派生プロトコルの開発、*BSD での XCAST6 実装、相互接続実験、WIDE 6Bone 上での試用を行ってきた。

2005 年度は、これまでの実績を元に以下の目標を掲げて研究開発活動を行った。

- XCAST プロトコル普及のためのプロモーション活動
- XCAST アプリケーションの使用環境向上
- オープンソースコミュニティとの連携
- IRTF/IETF での標準化に向けた活動
- XCAST 実験網運用

本報告書では 2005 年度の XCAST ワーキンググループの活動を以下の順に報告する。

1. XCAST プロトコルプロモーション活動
2. XCAST グループ管理システムの改良による使用環境の向上
3. XCAST 上の適用流量制御を付加したトランスポートプロトコルの開発
4. IETF/IRTF 活動
5. XCAST コミュニティーとの関係

第 2 章 XCAST プロトコルプロモーション活動

2.1 オンラインコミュニケーション

XCAST という新しい技術を世の中に広めていくためには、まず実際に使ってもらうことが重要である。そのために、IPv6 ネットワーク環境と容易なセットアップ方法を提供するため、一般ユーザーに「SoftEther」+「Windows Binary-kit」という 2 つのキーワードで XCAST 環境を構築してもらうスタンスを取った。

2.2 IPv6 の利用

XCAST の利用に関しては、IPv6 接続性のある環境での利用を想定している。しかしながら、現状では IPv6 接続性が提供されていないネットワーク環境も多い。XCAST を誰でも簡単に利用することが可能とするためには、IPv6 接続性のあるネットワークを利用できる環境を整えることも、XCAST のデプロイメントを行っていく上で重要である。我々は、一般ユーザーが IPv6 利用可能なネットワークに参加できるように、以下のような接続形態を用意した。

- VPN
- XCAST ルータ
- XCAST BOX

VPN に関しては、単体の PC で XCAST を利用したい場合に用いられる接続形態である。具体的には、名古屋大学に XCAST 利用のための公開セグメントを設け、「SoftEther」を利用して L2 接続を行った。このセグメントには RA が流されており、IPv4 NAT 配下からも IPv6 アドレスが取得可能であり、IPv6 接続性を確保することができる (図 2.1)。また、このセグメントには、登録されたユーザーのみが接続可能であり、IPv6 通信のみが許可されている。ここでの「登録されたユーザー」とは、SoftEther のアカウントを発行されたユーザーを指しており、XCAST の利用を目的とするならば誰でもアカウントを取得

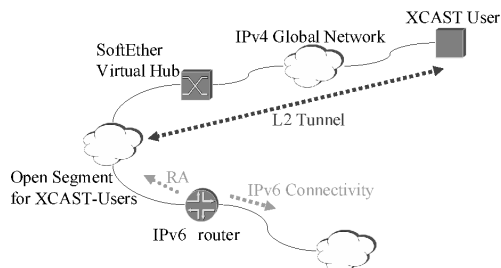


図 2.1. VPN による IPv6 接続性の確保



図 2.2. XCAST BOX

することができた。

XCAST ルータに関しては、あるネットワークの中の複数の PC で XCAST を利用したい場合に用いられる接続形態である。具体的には、次節で述べた方法で XCAST が利用可能なマシンを用意し、そこに IPv6 ルータとしての機能を持たせる。このルータは必要であれば DTCP (Dynamic Tunnel Configuration Protocol) を利用して IPv6 接続性を持つネットワークにトンネル接続を行う。XCAST を直接解釈することができない Windows マシンであっても XCAST ルータにおいて、X2U の機能を有効にし、配下に収容することによって XCAST を利用することが可能になった。

XCAST BOX は、XCAST ルータの機能を CF (Compact Flash) に書き込んだ HD レスのマシンであり、ネットワークケーブルをさし、簡単な初期設定を行うだけで、XCAST ルータとして利用可能なものである (図 2.2)。一時的な IPv6 ネットワークの構築にも有用であり、DTCP アカウントが必要ならば名古屋大学の DTCP サーバから発行可能である。

2.3 XCAST 環境の構築

多くの人に XCAST を利用してもらうためには、環境の構築が容易であることも重要である。現在、XCAST を利用することが可能であるプラットフォームとして、以下のものが挙げられる。

- Windows XP

- BSD (NetBSD、FreeBSD)
- XCAST6 Live CD

一般ユーザにとっては、Windows で XCAST を利用することがもっとも簡単な方法である。まず、IPv6 スタックをインストールしたマシンを用意する。そして、「Windows Binary-kit」インストーラをダウンロードし実行すると、XCAST が利用可能な環境ができあがる。ただし、Windows XP SP1 までの実装では、XCAST パケットを送信する際に raw socket を利用していたが、SP2 (Service Pack 2) 以降は仕様変更が行われ raw socket の利用ができなくなった。そのため、winpcap を利用して XCAST パケットを送信するように実装を修正した。また、IPv6 インターネット接続ファイアウォール (IPv6 ICF) などでは、デフォルト設定として IPv6 パケットをドロップする設定がなされていることが多いので、利用に際しては注意が必要である。

BSD 上で利用するためには、SourceForge にあるパッチを取得し、カーネルの再構築を行う。さらに、ライブラリや必要なツールのインストールを行うことで XCAST を利用することが可能になる。この方法は、これまでのところ最も動作実績が多く、安定した運用が報告されており、経験者向きである。

XCAST6 Live CD とは、BSD ベースで XCAST の利用環境が既に整えられたブート可能な CD のことであり、現在は「Ebifuryaa」と「FreeSBIE with XCAST」の 2 種類が用意されている。それぞれ、NetBSD、FreeBSD がベースとなっており、必要なライブラリやツールが 1 枚の CD にすべて収められている。これを利用することによって、環境構築の手間をかけずに簡単に XCAST を体験することが可能である。なお、Linux 上での利用のための開発も現在進められている最中である。

2.4 XCAST MATSURI

2.4.1 XCAST MATSURI の目的

我々はこれまでに、XCAST の普及活動を行うにあたり、SoftEther による IPv6 接続性の提供や、Windows バイナリの開発などを行ってきた。これらの利用環境が、実際に多くの人にとって使いやすいものであるのか、ということを検証するために「XCAST MATSURI」というイベントを開催した。また、このイベントは、「初めて利用するユーザが XCAST を体験してもらうきっかけ」とであると同時に、「初め



図 2.3. Max Connection Challenge!! の様子

て利用するために何が障壁であったかを、洗い出すための機会」, 「多人数でのオンラインコミュニケーションの利点と欠点を洗い出すための場」としても利用された。

さらに、多数の人間が一度にネットワークで会した場合、現実世界と同様に円滑なコミュニケーションが行えるかどうか実際に試し、オンラインコミュニケーションとリアルコミュニケーションの間にある差は何であるか、どのようなサポートツールが必要であるかの洗い出しを行った。

参加者には、イベント終了後にアンケートをとり、XCAST を利用するための手間や、実際に映像や音声を用いたコミュニケーションを行った感想などを答えてもらった。また、参加者の動向やネットワークの負荷を記録するために、ネットワークトラフィックやグループに参加したノードの観測を行い、客観的に MATSURI を評価する指標とした。さらに IRC のログや SoftEther 仮想 Hub のログ、参加者が保存したスクリーンショットを通して、MATSURI 活動の記録を行った。

2.4.2 XCAST MATSURI の様子

“XCAST MATSURI” は 2005 年 10 月 8 日から 10 月 10 日の 3 日間にかけて開催された。実際に参加者に集まってもらった時間は 3 日間で合計 7 時間であり、その間オンライン上で人が集まりコミュニケーションをはかっていた。MATSURI 全体を通して 3 つのグループが用意されており、それぞれパラ

レルで進行されていた。

それぞれのグループでは、初参加者のサポートを行うものや、テストストーリーミングを流し続けるもの、などがあった。また、XCAST を利用した BUG (BSD User Group) ミーティングの相互接続が行われ、“On Line (ネットワーク)” と “Real (現実)” の融合したコミュニケーションが展開された。さらに、同時に 1 つのグループに何人の参加者が集まることができるかを検証した「Max Connection Challenge!!」(図 2.3) も 10 月 9 日に行われた。

「Max Connection Challenge!!」は、技術的な面と社会的側面から重要な実験であった。技術的な面では、XCAST は実運用上、16 ノードが限界といわれているが、それを超えるとどうなるのか実際に試す意義深い機会であった。また、社会的な面では、それだけたくさんの人間が一堂に会すると、コミュニケーションは円滑に行うことができるのか検証する興味深い機会であった。

技術的には、最大で 30 ノードを同時に 1 つのグループに収容したという実績が作られたが、パケットのロスが激しく、映像や音声の品質の劣化が激しかった。これは、vic の送信レートをチューニングしていないなど、品質に関して考慮が行われなかったためである。XCAST の仕様上では、100 名を超えるメンバを収容することも可能である。

また、社会的には、30 ノードが集まったグループで話をするのは、困難がともなった。誰かの話に対して複数の人間が反応すると、誰が何を話しているの

か判断すること自体が大変であったが、ジェスチャーなど映像による情報の伝達でコミュニケーションを成立させることは可能であった。

さらに、MATSURI のために、SoftEther 仮想 Hub 専用マシンを用意した。新規アカウント申請が 40 件あり、最大同時接続数は 18 であった。18 人が同時に映像や音声の送受信を行うと、SoftEther の接続が安定しない状況も観測された。

2.4.3 XCAST MATSURI のサポート体制

今回は、MATSURI を運営するにあたり、参加者に対する告知には主に wiki を利用した。これは、多数の人間が協調して編集を行うことが容易であり、また、参加者が気軽に疑問点などを運営側と共有することが可能であったためである。

とくに、トラブルシューティングにおいて、wiki の効果が発揮されていた。誰かが操作などに関してうまくいかないことがあると、それを書き込み、それを見た同じトラブルを経験した人が解決策を示していく。このような蓄積が自然と行われることにより、後から参加したユーザにとって有益な FAQ (Frequently Asked Questions) が参加者の協調作業によってでき上がっていくことになった。

さらに、各グループに現在いくつのノードが参加しているかを wiki のリンクから確認することができるようにした。これによって、自分のノードが正しくグループに参加しているかどうかを確認したり、他のグループにどれだけ参加者がいるか調べたりすることができた。

また、MATSURI の告知に関しては、さまざまな団体の ML (Mailing List) を活用した。そこで興味を持っていた個人には “XCAST Live CD” の配布を行った。また、団体で利用したいと申し出があった拠点には “XCAST BOX” の提供と DTCP アカウントの発行を行い、/40 のアドレスブロック割り当てを行った。さらに、設定や利用方法に関して個別のサポートを行うなど人的・リソース的なバックアップを行った。

さらに、MATSURI 開催中には、参加者の間で情報の共有を行うために IRC を利用した。これを利用することで接続が不安定になった際の連絡や、設定方法のアドバイスなど、コミュニケーションの補助手段として有効に活用された。今回は IRC サーバとして、irc.nara.wide.ad.jp や irc.tokyo.wide.ad.jp

を利用し、MATSURI 用に開設されたチャンネルにおいて、連絡を取り合った。後から入ってきても議論の流れがわかるように、web にログがリアルタイムでアップされるようにした。

2.5 イベントの評価

2.5.1 トラフィックと参加状況

MATSURI 開催にともない、どれだけのトラフィックが流れるか、MATSURI ネットワークポロジのコアに存在するホストである xgate.xcast.jp において SNMP により観測を行った。このホストは WIDE X6-Bone に接続された DTCP サーバであり、全国の XCAST ルータのトラフィックは、このネットワークに收容される。つまり、ユーザ間の XCAST トラフィックは、このホストを経由することになるので、ここで観測されたトラフィックは、MATSURI 期間中のトラフィックとみなすことができる。

MATSURI の公式な開催時間は 1 日目が 18 時から 20 時の 2 時間、2 日目が 13 時から 16 時の 3 時間、3 日目が 18 時から 20 時の 2 時間である。図 2.4、図 2.5 の MRTG が示すように MATSURI 開催中はネットワークを流れるトラフィックが飛躍的に増加している。

「Max Connection Challenge!!」が行われた 10 月 9 日の 15 時 50 分前後に MATSURI 期間中のトラフィックのピークを記録した。瞬間最大トラフィックは、送信が 3465.9 KB/s、受信が 1535.5 KB/s であった。

また、MATSURI を客観的に評価するデータとして、各グループに参加しているノードの IPv6 アドレスを 1 分ごとに記録した。XCAST におけるメンバ管理システムである xcgroup ではクライアントからの “join” コマンドを受けるときに、グループメンバの IPv6 アドレスを保存したサーバ上のファイルを更新する。今回は、MATSURI で利用された xcgroup サーバである xmeet.xcast.jp において、1 分ごとにこのファイルのコピーを行い、MATSURI 期間中の参加ノード数の推移を記録した。

3 つのグループ (matsuri1、matsuri2、matsuri-test) に参加したノード数の推移は図 2.6 から図 2.8 のとおりである。このグラフは左から右へ時系列に沿った参加者数を示している。それぞれの瞬間最大参加ノード数は、matsuri1 が 30、matsuri2 が 17、matsuri-test が 13 であった。

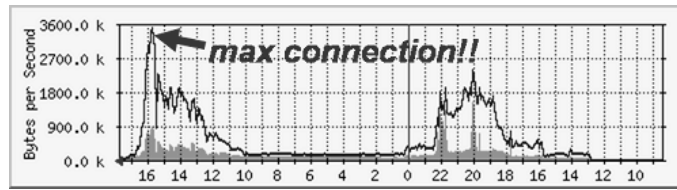


図 2.4. 1 日目から 2 日目のトラフィック

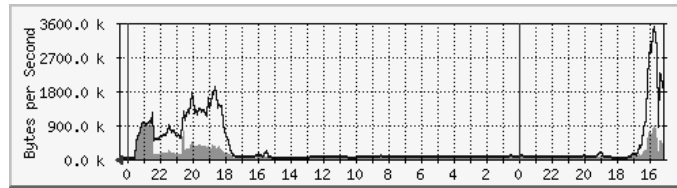


図 2.5. 2 日目から 3 日目のトラフィック

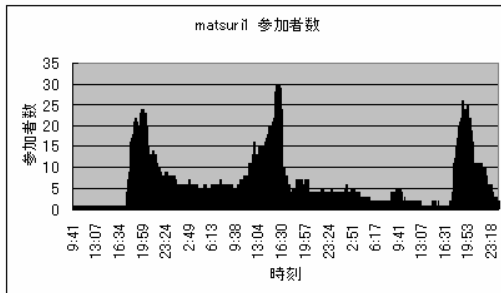


図 2.6. matsuri1 グループの参加者数

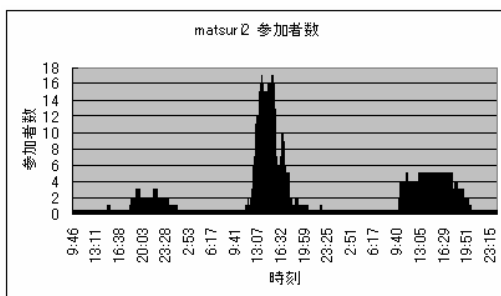


図 2.7. matsuri2 グループの参加者数

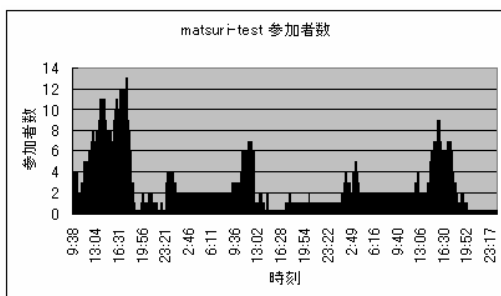


図 2.8. matsuri-test グループの参加者数

また、それぞれのユニークな参加ノード数は、matsuri1 が 88、matsuri2 が 45、matsuri-test が 58 であった。MATSURI 期間中のユニークな参加ノード数は全部で 142 であった。

2.5.2 アンケート結果

MATSURI 参加者の方に協力していただいた結果、35 人分の有効な結果が得られた。参加者が利用したプラットフォームとしては「WindowsXP SP2」(15 人)、「NetBSD」(11 人)、「FreeBSD」(8 人)、「WindowsXP SP1」(7 人)、「XCAST6 Live CD」(5 人) となった。また、IPv6 接続性に関しては、「SoftEther」(21 人)、「DTCP」(19 人)、「ネイティブ」(5 人)、「6to4」(1 人) となった。これらの結果から、「SoftEther」+「Windows Binary-kit」による XCSAT 環境の構築が、実際に多くの初参加者に利用されたことがわかった。

また、XCAST を使ったコミュニケーションに対する感想としては、「楽しい」(25 人)や「映像は必要である」(18 人)というものが多かったが、ストリームの品質に関しては、「画像や音声が届きにくかった」(7 人)や「他のツールと比べると少し落ちる」(5 人)といった意見もあった。参加者としては、映像のあるほうがコミュニケーションを円滑に行うことができるが、音声のほうがより重要であるので、音声の品質は最低限確保してほしいという、要望があることがわかった。

今回のイベントは、XCAST を体験してもらっかけという意味合いも持ち合わせていたが、地域別では、東海(19 人)、関東(12 人)、北海道(1 人)

信越(1人)、北陸(1人)、近畿(1人)、四国(1人)と全国各地からの参加者を集めることができた。

2.5.3 プロモーション活動の評価

XCAST MATSURI を通して、映像・音声によるオンラインコミュニケーションを今まで経験したことがなかった人が多数参加してくれたことにより、XCAST の現状を把握することができた。アンケートの結果から参加者の意見をまとめると、「実際に XCAST を使ってみると、顔をみながら話をできるのは面白い。しかし、導入の手間を考えると何らかのサポートが必要」となった。

今回は、チャットによるテクニカルサポート、wiki の情報を参考にする、わかる人に直接教えてもらう、サポート窓口でメールする、といった方法をとった参加者が多かった。このようなサポートを評価してくれた参加者も多かったため、今後も継続的にサポートを行う予定である。

2.5.4 まとめ

今回のイベントで最も大きな反響があったのは WindowsXP SP2 での XCAST 利用である。これまで、あまり多くの Windows ユーザがいなかったため、対応も遅れがちであったが、初参加の方々の要望の多さに応え、SP2 対応バイナリをリリースすることができた。もっと早い時期に対応していれば、よりたくさんの方がデータを送り出すことができ、MATSURI の雰囲気も変わっていたかもしれない。今後も多様な環境においてユーザが、XCAST を使ったビジュアルコミュニケーションを利用することができるように環境の整備やツールの充実をはかり、XCAST を普及させていきたい。XCAST に興味を持たれた方や、疑問等がある方は、xcast@el.itc.nagoya-u.ac.jp までご連絡頂きたい。

第 3 章 XCAST グループ管理システムの改良による使用環境の向上

3.1 導入(XCAST を利用した統合コミュニケーション環境の構築)

3.1.1 背景と目的

現在では、PC 上でもテキスト/ボイス/ビデオ

チャットなどのリアルタイムコミュニケーションが一般化してきている。文字の持つ明確な伝達性および音声や画像の持つ言語外の表現の有用性を考慮すると、これらのコミュニケーション手段は統合的に使えるのが望ましい。このような統合コミュニケーション環境はいくつか存在しているが、音声や映像などを扱った場合、グループメンバー数の制限が厳しかったり、グループ数や全ユーザ数が増加した場合のスケーラビリティに問題がある。組織内での利用だけでなく、一般ユーザが広く利用できるシステムでは、グループ数や全ユーザ数に対するスケーラビリティは重要な要素である。本研究では、グループ数や全ユーザ数に対するスケーラビリティを持ちテキスト/ボイス/ビデオチャットが利用できる統合コミュニケーション環境を構築することを目的とする。

3.1.2 要求事項

本研究が目指している、グループ数や全ユーザ数に対して高いスケーラビリティを持った統合コミュニケーション環境を構築するための要求事項を以下に列挙する。

1. ネットワーク帯域の消費が大きいボイス/ビデオチャットを行った場合でも、グループ数と全ユーザ数の増加に影響されない通信モデルや通信方式が必要である。
2. 全体のユーザ数の増加に対応し、メンバの管理や認証などを適切に行う必要がある。
3. 同時に複数人とのグループコミュニケーションを行う場合、コミュニケーションを行う場の作成や管理、さらに部屋の公開/非公開やユーザの権限設定などのさまざまなグループマネジメント機能を補う必要がある。
4. グループコミュニケーションにおいては、テキストチャットだけでなく、プレゼンス情報の交換、他人を部屋に招待するなどの標準的な IM (Instant Messaging) 機能が必要である。
5. ユーザがコミュニケーションを行う際、無数に存在する部屋の中から、自分が希望する部屋を容易に検索するためには、グループに参加する前に、現在どのようなグループがあり、どのようなユーザが存在するのかわかることができる機能が必要である。

そこで本研究では、要求事項の 1 を解決するため、サーバを介す通信モデルではサーバに負荷がかかるこ

とから、サーバを介さずクライアント同士通信をするモデルを採用し、さらに、クライアント側ネットワーク帯域を軽減するために、比較的少人数のグループを多数扱うことに適したプロトコルである XCAST (eXplicit Multi-Unicast) を利用したビデオ会議システムを用いることとした。一方、要求事項の 2、3、4、5 を解決するため、XML ベースのオープンソース IM の Jabber を用いることとした。XCAST を利用したビデオ会議システムと Jabber を組み合わせることで、要求事項をある程度満たすシステムを構築することはできるが、XCAST を利用したビデオ会議システムのグループマネジメントシステムでは、XCAST のスケーラビリティによって獲得されるグループとそれに対応した全ユーザをうまくマネジメントできていない。本研究が目指している高いスケーラビリティを持ったシステムを完成させるためには、スケーラビリティによって獲得されたユーザを適切にマネジメントする機能を用意し、スケーラビリティを有効に活用する必要があると考えられる。そこで本研究では、Jabber を拡張することにより、Jabber を新たなグループマネジメントシステムとして置き換え、グループ数や全ユーザ数に対して高いスケーラビリティを持ったシステムを構築した。

3.2 既存のシステムによる実現

前節で紹介したシステムを組み合わせることで、以下の要求を満たすシステムを構築することができる。

- テキストチャット / ボイスチャット / ビデオチャット
- 不特定多数のユーザの参加
- 不特定多数のグループの生成

XCAST とそのグループマネジメントシステムの xcgroup、そしてボイス / ビデオチャットアプリケーションである vic/rat から構成されるビデオ会議システムと Jabber を組み合わせることで実現している。画像と音声のようなデータ量の大きいものは、サーバを介さず XCAST によりクライアント同士で直接通信をし、テキストチャットのようなデータ量の小さいものは、サーバを介して通信することにしている。これにより、グループ数やそれに対する全ユーザ数が増加した場合でも、テキストチャットとボイス / ビデオチャットが統合的に扱えるようになる。

3.3 問題点とその解決

単に組み合わせたシステムでは、XCAST のスケーラビリティによって獲得されるグループやそれに対応するすべてのユーザをうまく管理できていない。スケーラビリティを持ったシステムを構築するためには、次に列挙するような機能を設け、スケーラビリティを有効に活用する機能が必要であると考えられる。

認証機能 不特定多数のユーザが存在する場合、すべてのユーザに対して、適切に認証が行われる必要があると考えられる。

グループ情報取得機能 ユーザが増加した場合、それに比例してさまざまな用途を持ったグループが生成される。ユーザは無数に存在するグループの中から、自分の参加したいグループを適切に探し出すことで、グループの参加が容易になる。よって、グループに参加する前に、適切にグループ情報を取得できるような機能が必要であると考えられる。

メンバ情報取得機能 グループ情報取得機能と同様、ユーザ数が増加した場合、さまざまなユーザが存在することになる。参加者はグループに参加する前に、現在のグループ内メンバを把握することができれば、グループの参加が容易になる。よって、グループに参加する前に、適切にメンバ情報を取得できるような機能が必要であると考えられる。

招待機能 ユーザ数に対するスケーラビリティを活用するためには、ユーザ同士の間で気軽にメッセージを送り合い、グループを拡大していかなければならない。よって、他の参加者を容易にグループに参加させるために、他人を招待する機能が必要であると考えられる。

部屋の属性制御機能 ユーザ数が増加した場合のコミュニケーションでは、ユーザの権限設定やユーザが参加している部屋の公開 / 非公開などの詳細な属性制御が必要とされる。複数人のコミュニケーションでは、このような機能が有効であると考えられる。

本研究では、XCAST を利用したビデオ会議システムに上記の機能を持たせることにより、XCAST のスケーラビリティを有効に活用できると考えた。一方、XCAST を利用したビデオ会議システムのグルー

ブマネジメントシステムである xcgroup の機能は、グループへの IP アドレスの登録 / 削除機能、IP アドレスリスト取得機能のみである。そこで本研究では、Jabber に xcgroup の機能を持たせることにより xcgroup を排除し、Jabber を新たなグループマネジメントシステムとして置き換えるという構造を考えた。

このために以下のような変更を行った。

- Jabber のグループ情報にボイス / ビデオチャット用の属性を追加した。
- XMPP メッセージに拡張を施してクライアントの IP アドレスを格納し、Jabber サーバにメンバの IP アドレスを登録するようにした。
- XMPP メッセージを拡張してボイス / ビデオチャットへの招待機能を実装し、友達リストのポップアップメニューに追加することで、Jabber クライアントと XCAST を利用したビデオ会議システムの統合を行った。

これにより以下が可能となった。

- Jabber が持っている認証機能やグループ参加時のメンバ制限機能を用いて、不特定多数のユーザが存在する場合でも、適切な認証機能を持つことができた。
- Jabber が持っているグループ情報取得機能を用いて、サーバにどのようなグループが存在するか、あるグループでボイス / ビデオチャットが行えるか、そのグループに現在誰が参加しているかなどの情報を事前に取得できるようになり、無数に存在するグループ情報やユーザ情報を適切に取得でき、よりユーザが参加し易い環境となった。
- 他のユーザに対して、グループへの招待メッセージが送信可能となり、他のユーザもグループに参加し易くなった。

以上のことが可能となり、XCAST のスケーラビリティによって獲得されるユーザをうまくマネジメントできるようになったと考えられる。これにより、グループ数や全ユーザ数に対して高いスケーラビリティを持ったシステムを完成させることができたと考えられる。

3.4 研究成果と新たな研究テーマ

本研究では、グループ数とそれに対する全ユーザ数に対して、スケーラビリティのある統合コミュニ

ケーション環境を構築するために、XCAST を利用した。そして、XCAST を有効に活用したスケーラビリティの高いシステムを構築するために、XCAST 通信を行うためのグループマネジメントシステムである xcgroup を、オープンソース IM である Jabber を拡張することで、新たなグループマネジメントシステムとして置き換えた。今後の課題としては、クライアント側にまだ IP アドレスリストの登録 / 取得機能が実装されていないので、サーバ側と関係が取れていない。これを解決することが今後の課題の 1 つである。また、XCAST のビデオ会議システムには部屋の属性制御機能が欠如している。この機能を補うことも今後の課題である。そして最大の課題は、本研究で構築したシステムを実際に多グループやそれに対応するすべてのユーザに対して利用し、その定量的な評価を取り、システムの有用性を検証することである。

今回利用した Jabber は XML ベースのオープンソース IM であり、拡張性の高い独自のプロトコルを規定している。しかし、実際に拡張を行った経験から、追加機能の拡張があまり容易ではないと感じた。また、世の中には様々な嗜好を持ったグループが存在し、ネットワーク上でも「濃い」コミュニケーションが行われている。それらのグループにはそれぞれグループ特有の要求があり、グループごとに欲しい機能は多種多様である。その上ユーザは要求に対して貪欲であるため、1 つの要求を満たしても別の要求が際限なく繰り返す。しかし、既存のコミュニケーションツールでは、変化するユーザの要求に迅速にまた際限なく応えることは困難であると考えられる。

このような背景から、要求に迅速にかつ柔軟に対応できるシステムを構築することができれば、拡張機能を容易に追加することができるようになり、多種多様で際限ない要求に対応できると考えられる。このような要求に迅速にかつ柔軟に対応できるシステムを構築するため、MVC2 をベースとした拡張が容易で開発効率の高い Web アプリケーション用フレームワーク (Ruby on rails) を利用することとした。

しかし、現在良く語られている Web アプリケーションにおける MVC アーキテクチャは、PULL 型のモデルを前提としている。これをグループコミュニケーションに適応した場合、同期的な通信を扱うことはできるが、非同期的な通信を扱うことができないという

問題を抱えている。この問題は、Subscribe/Notify 構造を導入することで解決することができるが、フレームワークレベルで、非同期通信に対応した上での API の定義などが必要となる。これが今後の研究のテーマとなる。

第 4 章 XCAST 上での適用流量制御を付加したトランスポートプロトコルの開発

XCAST 上で、輻輳制御を行う技術 SICC (Sender Initiated Congestion Control) の開発評価を行った。XCAST および SICC をネットワークカメラに適用した試作機を 2005 年秋合宿および IC2005 で展示した。本章では、これらの活動を以下の 3 つに分けて報告する。

- 2005 年秋合宿での、XCAST6 対応ネットワークカメラの展示
- IC2005 での、XCAST6/SICC 対応ネットワークカメラの展示
- XCAST6 上での SICC の NS-2 による性能評価

4.1 2005 年秋合宿での、XCAST6 対応ネットワークカメラの展示

IPv6 上の代表的な SGM (Small Group Multicast) 方式である XCAST6 を Panasonic 製ネットワークカメラに適用した。本デモでは実際のインターネット環境に近い状況下 (帯域・遅延) での 2 本の TCP フローと 1 本の XCAST6 フローの違いを目で見て分かる形で再現する。本デモシステムを 2005 年

秋合宿で展示した。

なお、本展示で得られた意見などより詳しい情報は、wide-memo-xcast-2005autum-camera-demo-00.txt にまとめてあるので参照いただきたい。

4.2 IC2005 での、XCAST6/SICC 対応ネットワークカメラの展示

4.2.1 はじめに

ネットワークカメラは CCD センサーや内蔵マイクを標準搭載し、コミュニケーションデバイスとして利用可能である。しかし、通信プロトコルには HTTP を利用しているため同時接続数に限りがあり、同報配信やグループ通信には適していない。また、グループ通信では、既存ネットワークへの影響を最小限に抑え、かつ、送受信者間で許容され得る最大限の送信速度で情報を配送できることが望ましい。そこで、我々はこれらの要件を満たす技術として XCAST6 および SICC を採用し、IPv6 ネットワークカメラへ実装した。

4.2.2 内部設計

本節では、XCAST6/SICC 対応 IPv6 ネットワークカメラの内部設計について記す。図 4.1 はその内部構成である。

XCAST6 スタック

我々は IPv6 ネットワークカメラを XCAST6 送信ノードとして利用することを想定し、XCAST6 送信機能のみを実装した。グループメンバー情報はネットワークカメラの持つ設定インタフェースから直接指定する。

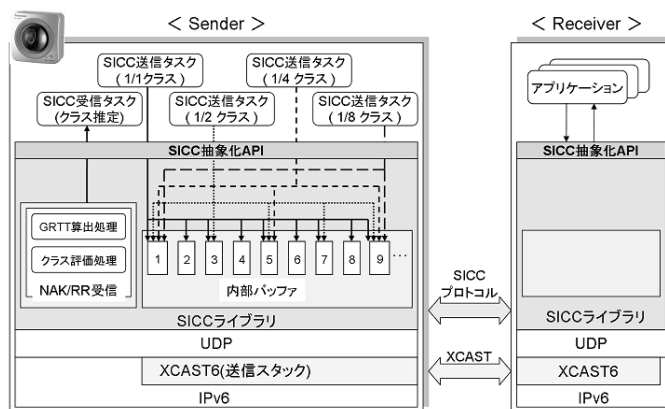


図 4.1. XCAST6/SICC 対応ネットワークカメラ内部構成

SICC ライブラリ

SICC ライブラリは SICC 機能を抽象化する API を提供する。SICC ライブラリは JPEG ドライバからフレームを取得し、一度内部バッファにキューイングする。これは、後述する SICC 送信タスクが送信レートに従いフレームを間引きする必要があるためである

SICC 送受信タスク

XCAST6/SICC 対応ネットワークカメラは、送信レートクラスごとに送信タスクを生成する。送信レートは、1/1 クラス、1/2 クラス、1/4 クラス、1/8 クラスの 4 クラス用意した。各送信タスクは抽象化 API を通じ SICC ライブラリにバッファリングされた JPEG フレームのうち、送信すべきフレームのポインタ情報を持ち、ポインタ先のフレームを XCAST6 スタックへ送る。

SICC 受信タスクは、SICC 受信メンバーからの RR (Receiver Report) および NAK (Negative Acknowledgement) フィードバックを処理し、GRTT (Greatest Round Trip Time) にもとづき送信レートを推定しメンバーを適切なクラスに収容する。

4.2.3 デモ環境

本デモ環境のシステム構成およびコンポーネントは、図 4.2 の通りである。中継経路上に Dummynet

を配置し、狭帯域リンクをエミュレートする。狭帯域リンク上には XCAST6/SICC 対応ノードと非対応ノードが接続され、XCAST パケットと TCP フローが共存する。

なお、4 章の内容は、wide-paper-xcast-sato-ic2005-00.txt に記載の情報から参照可能である。

4.3 XCAST6 上での SICC の NS-2 による性能評価

4.3.1 Introduction

We have previously proposed a congestion control method called Sender Initiated Congestion Control (SICC) that is intended to provide TCP Fairness, Fast Congestion Avoidance and Intra Session Fairness for applications based on Small Group Multicast (SGM). SGM is a suitable method to transmit a packet to a group containing from 10 to 100 participants.

In SGM, the sender specifies receivers' addresses in a packet explicitly. XCAST6 is the most typical mechanism based on SGM in IPv6 network. The XCAST aware router copies and forwards packets using the unicast routing table thus that no control packet is needed to exchange multicast routing entries. In addition, XCAST6 has a tunneling mechanism which can be used to provide connectivity between XCAST aware nodes

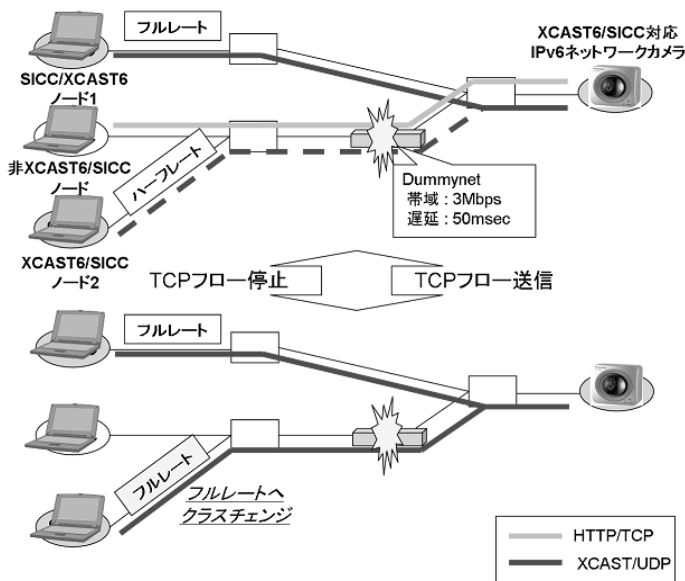


図 4.2. XCAST6/SICC デモ環境

(i.e., router or receiver) and to pass over the non-XCAST aware routers. As a result, it is possible to gradually deploy XCAST6 over the Internet. Since XCAST6 allows packets to be forwarded among receivers even if there is no XCAST aware router along the forwarding path, the performance of mechanisms based on XCAST6 are significantly affected by the arrangement of XCAST aware router on the networks.

We evaluate the performance of SICC, which is currently integrated into XCAST6, using the network simulator ns-2.

4.4 SICC Protocol

4.4.1 Target Application

The target Application of SICC is real-time streaming which consists of a selectable sequence of frames (e.g. Motion-JPEG and DV).

4.4.2 SICC feature

SICC integrates the following functions into SGM.

- **TCP Fairness and Fast Congestion Control**

In the Internet, it is important to avoid congestion and maintain fairness with other competing best-effort traffic. A transport protocol should avoid any congestion which arises on the path from a sender to receivers.

When we design such a transport protocol, it is sufficient to consider fairness of the TCP traffic (TCP Fairness) which is the major traffic in the current Internet. Because a large number of group communication sessions will be held simultaneously competing with other best-effort traffic, TCP Fairness and Congestion Control are very important functions.

- **Intra Session Fairness**

It is preferable that a sender can transmit to each receiver in various environments at the allowed maximum transmitting rate in the multicast group communication. This is called Intra Session Fairness which is a requirement specific to multicast transport

protocol. We should consider the Intra Session Fairness in designing a multicast congestion control method.

4.4.3 Protocol Overview

- **SICC CLASS and Transmitting SGM Packet**

The SICC sender has several CLASSes which transmits packets at a constant bit rate. The sender classifies receivers into the suitable CLASS based on estimation of acceptable sending rate, and each CLASS transmits SGM packet that specifies the addresses of the classified receivers.

The number of CLASSes is statically fixed, that is, the user of SICC configures the number of CLASSes when he opens the SICC socket at the sender node.

Figure 4.3 shows an example of the above mentioned CLASS (e.g., C1, C2, C3). The transmitting rate of each CLASS is defined as one-half of its upper CLASS. This can be achieved by thinning out or sampling picture elements by the SICC stack. The sender first classifies the receivers (R1, R2, R3, R4) into the corresponding CLASSes based on their acceptable sending rates, and then transmits SGM packets to the receivers in each CLASS.

- **Receiver Feedback**

Each receiver regularly feeds back a report to the sender. The interval of feedback is different in each CLASS, and uses the maximum RTT in the receivers in each CLASS. Additionally, when a receiver detects the packet drop indicating competition with other traffic or congestion, the receiver feeds back a report immediately to the sender. Afterwards, the sender estimates the acceptable sending rate of the receiver based on the feedback, and classifies the receiver into the suitable CLASS. As a result, the transmitting rate to the receiver is changed to be the best of acceptable ones.

This is the reason why SICC achieves fast

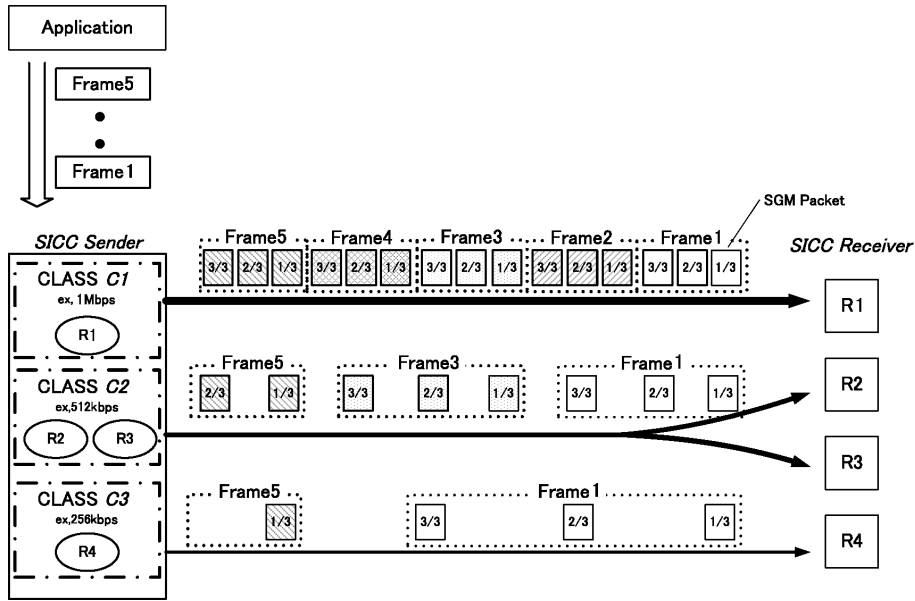


Fig. 4.3. SICC CLASS

congestion avoidance and the Intra Session Fairness.

– Classification based on TFRC Equation

To estimate the rate that each receiver can accept, SICC adopted the TCP-friendly Rate Control protocol (TFRC) approach. TFRC is an equation base congestion control method for unicast applications which provides smooth rate control while maintaining fairness with other TCP traffic competing with the flow.

SICC sender estimates the acceptable sending rate of receivers based on the Equation (1) with the feedback from each receivers. In the Equation (1), s [byte] is the packet size, R [sec] is the estimated RTT tempered with past history by weighted average, p is the loss event rate and t_RTO [sec] is the TCP retransmission timeout value (usually $4 * R$). After the estimation, the SICC sender classifies receivers into the suitable CLASS. In SICC, the sender has several CLASSes: C_i which transmits packets to accommodated receivers using SGM at constant bit rate B_i [bps]. The sender classifies the receiver to the CLASS: C_x so that the following Equation (2) is satisfied.

Therefore, SICC can control the rate while maintaining fairness with the TCP traffic in the routes to the receivers.

$$X_{cal} = \frac{8s}{R(\sqrt{2p/3} + t_RTO * \sqrt{3p/8 * p * (1 + 32p^2)})} \quad (1)$$

$$B_{x+1} < B_x \leq X_{cal} < B_{x-1} \quad (2)$$

4.4.4 Confirming Basic Protocol Feature

We have evaluated TCP Fairness and Intra Session Fairness which are the main characteristics of SICC. This section shows the results of our evaluation using the network simulator (ns-2).

4.4.5 TCP Fairness

To achieve TCP Fairness, SICC adopts TFRC approach while estimating the sending rate. We confirm whether SICC applied on XCAST6 is able to share the bandwidth with the TCP flows fairly on the bottleneck link.

First of all, we use the well known single-bottleneck link topology (Figure 4.4) to evaluate whether each flow maintains fairness with other flows when both the SICC flows and TCP flows increase synchronously.

Figure 4.5 and Figure 4.6 show the throughput of a SICC flow and a TCP (Reno) flow competing

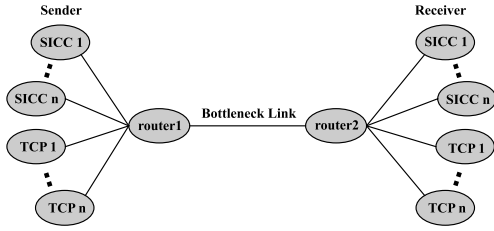


Fig. 4.4. Single-bottleneck topology

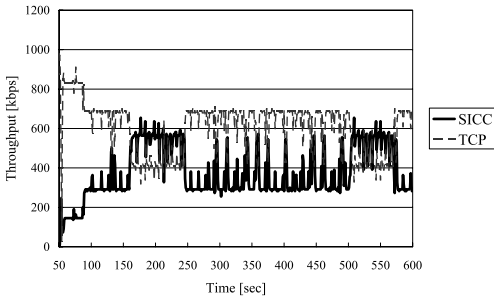


Fig. 4.5. SICC and TCP throughput (DropTail)

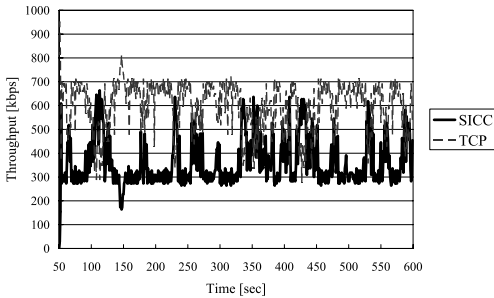


Fig. 4.6. SICC and TCP throughput (RED)

with SICC. In this simulation, the number of CLASSES is five, and the transmitting rate of each CLASS is {C1: 1 Mbps, C2: 512 kbps, C3: 256 kbps, C4: 128 kbps, C5: 64 kbps}. In the bottleneck link, bandwidth is 1 Mbps, delay is 64 ms, and DropTail or RED queue. These results show that SICC can perform rate control keeping fairness with TCP flows.

We introduce the index: F that shows whether SICC and co-existing TCP flows can fairly share the bandwidth on the bottleneck (3). In this equation, $linkbw$ [bps] is the bandwidth of bottleneck link, i is the number of flows on the bottleneck link, $thput_avg$ [bps] is the average throughput of each flow. F becomes 1 when the flow fairly shares bandwidth of the link.

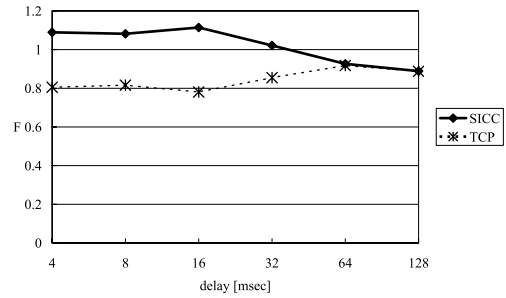


Fig. 4.7. TCP Fairness with n SICC and n TCP flows

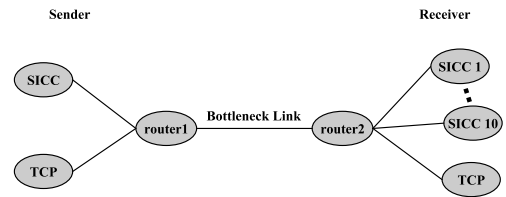


Fig. 4.8. Single-bottleneck topology2

$$F = \frac{thput_avg}{\frac{linkbw}{i}} \quad (3)$$

Figure 4.7 illustrates the fairness of each flow, when n is changed 1, 2, 4, 8, 16, 32, 64, 128 on the simulator. The simulation time is 600 seconds, the delay is 64 ms, the queue is RED, bandwidth and queue length of the bottleneck link are increased in proportion to the number of flows. The CLASS transmitting rate of SICC is same as Figure 4.5 and 4.6.

It shows the fairness (F) of each SICC flow indicates the value of about 0.75 to 1, the one of each TCP flow indicate the value from 0.9 to 1.2. According to the above results, even if the number of both flows increases, the average throughput of all the competing flows is within twice in each other. Therefore it is confirmed each SICC flow can share the bandwidth with other SICC flows and TCP flows fairly on a bottleneck link.

Next, we evaluate whether a SICC flow and a TCP flow maintains fairness with each other on a bottleneck link when the SICC sender transmits XCAST6 packets to ten receivers in the topology shown in Figure 4.8.

Figure 4.9 illustrates the fairness of each flow, when delay is changed 4, 8, 16, 32, 64, 128 on the simulator. The simulation time is 600 seconds, the

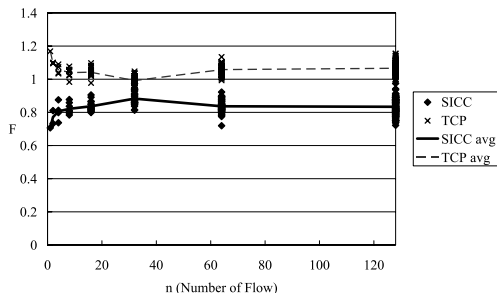


Fig. 4.9. TCP Fairness with SICC multicast and TCP

bandwidth is 1 Mbps. The CLASS transmitting rate of SICC is same as Figure 4.5 and 4.6.

It shows the fairness (F) of a SICC flow indicates the value of about 0.9 to 1.1, the one of a TCP flow indicate the value from 0.8 to 0.9. Therefore it is confirmed multicast communication on SICC can share the bandwidth with TCP flow fairly on a bottleneck link.

4.4.6 Intra Session Fairness

To confirm Intra Session Fairness of SICC, we conduct an evaluation using a simple multicast topology as shown in Figure 4.10, where there are ten receivers and one sender. The bandwidth of

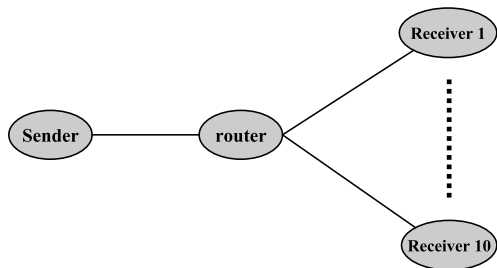


Fig. 4.10. Simple Multicast topology

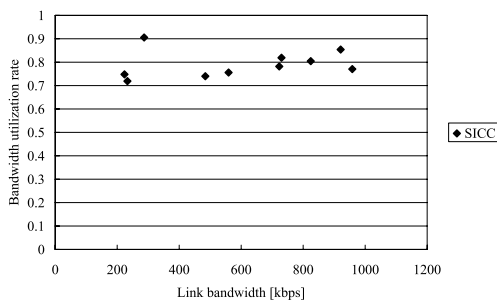


Fig. 4.11. Bandwidth Using Rate

the link between the router and each receiver is generated randomly from a uniform distribution between 64 kbps to 1 Mbps, and the link delay is generated randomly from a uniform distribution between 4 ms to 128 ms. In addition, we set the number of the CLASSes is five, the transmitting rate of each CLASS is {C1: 1 Mbps, C2: 512 kbps, C3: 256 kbps, C4: 128 kbps, C5: 64 kbps} and the simulation time is 600 seconds.

Figure 4.11 shows the relative bandwidth utilization of each receiver to various types of available uplink bandwidths. We observe in Figure 4.11 that each receiver is capable of receiving packets at the rate corresponding to available uplink bandwidth. This shows that SICC achieves its objective of Intra Session Fairness.

More detail information can be extract from wide-paper-xcast-yoneda-ic2005-00.txt.

4.5 IETF/IRTF 活動

我々は、グループ通信を実現する通信技術として XCAST/SICC 等の IETF/IRTF での標準化を計画している。また、XCAST の普及に向けた取り組みとして、Linux、*BSD などへの XCAST プロトコルの導入を目指した実験番号取得の活動を展開している。これらの活動を表した以下の文書ごとに目的と活動の進捗を報告する。

- 実験番号申請の草案
- XCAST 基本仕様の Experimental RFC 化に向けた草案
- XCAST コミュニティの活動を示した草案

さらに、我々は、同時に IRTF での標準化準備活動を行っている。この活動についても報告する。

4.6 実験番号申請の草案

XCAST6 は、グループ通信の実現する実験的なプロトコルである。XCAST6 を広域なインターネット上で実験運用するあたり、不要なトラブルを防ぐため、XCAST6 が用いるオプションヘッダ等の実験番号空間が確保されることが望ましい。このため、我々は、実験番号の要求をまとめた草案を投稿した。本草案は、現在、Bill Fenner がレビュー中である。これらの実験番号空間が割り振られれば、XCAST6 などの実験プロトコルを Linux や *BSD に導入し、これ

らを用いた広域実験を安全に行うことが可能となる。

なお、本草案の本文は、draft-hsu-xcast-iana-considerations-00.txt を参照いただきたい。

4.7 XCAST 基本仕様の Experimental RFC 化に向けた草案

XCAST の基本仕様を規定した草案を Experimental RFC に向けた草案として RFC Editor に投稿した。現在、RFC Editor による 2 度のレビューを受け、修正版を再投稿した段階である。なお、本草案の本文は、draft-ooms-xcast-basic-spec-09.txt を参照いただきたい。

4.7.1 XCAST コミュニティの活動を示した草案

XCAST 関連プロトコルの標準化を開始するにあたり、XCAST コミュニティの活動状態を示した草案を投稿した。本草案の本文は、draft-hsu-xcast-bcp-2004-01.txt を参照いただきたい。

4.7.2 IRTF での標準化準備活動

XCAST は、2000 年ごろから IETF での標準化活動を行っている。2002 年の IETF 横浜で IDMR (Inter Domain Multicast Routing) の設計チームが解散したことに代表されるように、広域で同時に多数の通信が発生するような通信に IP multicast を適用することを疑問視する見解も見受けられるようになってきた。(例：RFC2902[43])

我々は、このような現実を受け、XCAST や ALM (Application Level Multicast) などグループ通信に適したマルチキャスト方式の有効範囲を整理する目的で IRTF-RG の設立に向けた取り組みを展開している。現在までに、IRTF Chair である Aaron Falk、P2P-RG の Chair である Bobby Bhattacharjee らと会議を重ね、第 65 回 IETF で非公式な会議を行うことで合意している。

第 5 章 XCAST コミュニティとの関係

5.1 X6-Bone

2002 年度から運用を行っている仮想的 XCAST6 ネットワーク X6-Bone は、半透過トンネル手法に

頼らない、XCAST6 ルーター間を直接結ぶネットワーク上で、各種実験を行う目的で設置している。ネットワークポロジは慶応大学相南藤沢キャンパス内に設置した XCAST6/NetBSD PC ルーター (xgate.xcast.jp) をハブとし、協力団体を DTCP による v6/v4 トンネルで接続するスター型としている。協力団体は昨年度より 2 件増加した。

- 越後 BSD User Club (EBUG)
- AI3 (Asian Internet Infrastructure Initiative) における XCAST ワーキンググループより USM (University of Science Malaysia)

一方で、3ffe: を prefix とする、いわゆる WIDE6Bone のアドレスの利用停止が 2006 年 6 月に計画されている。X6-Bone は、XCAST6 が用いている Hob by Hop オプションを通過させないルーターが存在しても XCAST6 の実験を行うことを可能とするため、運用を継続する計画である。

5.1.1 オープンソースコミュニティとの協調

WIDE プロジェクトで作成した XCAST6 成果物は、BSD 形式のライセンスによるオープンソースとして、sourceforge.net を通じて配布してきた。これにより、WIDE 外部のオープンソースコミュニティの手で各種の派生プログラムが作成された。XCAST ワーキンググループは、これらのコミュニティ内で緩やかな連携を構成する目的で、ML の運営、X6-Bone への収容、ミーティング実験などで、成果を共有しあう場を提供した。

5.1.2 BSD 版 XCAST6-kit 改版

2004 年度より、NetBSD、FreeBSD の改版に対する xcast6-kit の追従のための update が WIDE 外で行われるようになった。これが一貫して継続的に開発されていき、各 BSD コミュニティのユーザに便利に使用してもらおう状況を作る事を目的に、主要な開発者に対して XCAST6-kit の正式な developer として活動していただくよう依頼し、以下の 2 名に承諾をいただいた。

- 梅本肇氏 (CBUG、FreeBSD committer)
- 濱嶋克臣氏 (NBUG、NetBSD committer)

両氏により、FreeBSD 6.0、NetBSD 3.0 への対応をリリースと同時に迅速に行われている。

5.1.3 Linux 版 XCAST kit

2005 年度に Linux 2.6 ベースの XCAST カーネルの安定化作業が行われた。2.6.10 カーネルでのテストが行われ、送受動作、転送動作とも安定稼動が確認できている。アプリケーションについては、vic の動作報告が行われている。rat については、アプリケーションでの音声デバイスの認識の問題が報告されている。

付録 X6Bone 協力組織

参加団体	割当 prefix
CBUG (調布 / 西東京 *BSD ユーザーグループ)	3ffe:051b:4400::/40
SBUG (Shikoku *BSDUsers Group)	3ffe:051b:4500::/40
NoBUG (北海道 *BSD ユーザーズグループ)	3ffe:051b:4600::/40
BAY-BSD (FreeBSD と飲もう会・BAY-BSD ユーザーズグループ)	3ffe:051b:4700::/40
NBUG (名古屋 *BSD ユーザーズグループ)	3ffe:051b:4900::/40
GPUG (西日本 XCAST Protocol Users Group)	3ffe:051b:4a00::/40
IRISA (irisa.fr)	3ffe:051b:4b00::/40
APLHA (KNOPPIX w/ XCAST 実験グループ)	3ffe:051b:400f::/48
SPUG (静岡 XCAST Protocol Users Group)	3ffe:051b:4105::/48
EBUG (越後 *BSD ユーザーズグループ)	3ffe:051b:4112::/48
USM (University of Science Malaysia)	3ffe:051b:4111::/48