修 士 論 文

実用的な高信頼マルチキャストに関する研究

指導教官　篠田陽一 助教授

北陸先端科学技術大学院大学
情報科学研究科情報システム学専攻

村本 衛一

平成 12 年 3 月 1 日

目 次

図 目 次

表 目 次

第 1 章

はじめに

1.1　本研究の背景

1.1.1　1 対多、多対多の要求

当初、研究者の間で『つながること』を目的に発展してきたインターネットは世界規模の情報流通基盤になりつつある。日本でも、ネットワークを通じた証券取引の手数料の自由化や音楽など著作物販売に関する規制緩和などインターネットの利用を前提とした施策が進められている。これらの動きは産業構造の変革を促し、さらにインターネット通信に対して新たな要求を生み出だす。

例えば、インターネットを利用した証券取引では迅速な株価情報が重要である。現在、この提供はその伝送量を抑えるために顧客ごとに登録された銘柄の情報だけを配信するようにしている。それでも、この仕組みは、1 対 1 の通信を前提としているため顧客の増加とともにサーバ負荷が高め、インターネット上での通信量を増大させてしまう。

音楽の配信サービスについても、一般ユーザがユニキャスト通信を用いて自らが希望する音楽データをダウンロードする方式が採用されている。これでは、例えば数万人が同時に欲しがるようなデータを快適に顧客提供するためには、大容量回線と強力なサーバという膨大な設備投資が必要になる。このれらの問題は 1 対多、多対多の通信を行なうマルチキャストを用いて、契約した顧客に対してサーバから対象データを一斉同報する方式を採用すれば解決する。

また、近年、インターネット上で生放送によるビデオ、オーディオの情報発信など実時間性を必要とする同報通信も増えてきている。これらを高品質にスムースにかつ低コストで配送する必要性が高まっている。

1.1.2 信頼性を提供しないIP マルチキャスト

１対多、多対多の通信を実現する第3層の技術として、IP マルチキャストがあげられる。IP マルチキャストは最善努力型の転送を行なうのみで配送の信頼性は提供しない。したがって現在では画像や音声の配信など、多少のパケット落ちなどが許容される通信に用いられてきている。配送の信頼性向上させるためには、ネットワーク層より上の層で再送、冗長符合化などの機構を用いる必要がある。

1.1.3 高信頼マルチキャストの研究

高信頼マルチキャストは、インターネット技術の延長上で、1対多、多対多の通信にユニキャストで用いられているTCP(Transport Control Protocol)のような信頼性を提供するための技術である。現在、損失回復の方法に特徴をもつ様々な高信頼マルチキャストプロトコルが提案されている。また、マルチキャストアプリケーションには多彩な要求が存在するため、ただ一つのプロトコルを標準化することは難しいとされている[9]。

1.2 本研究の目的

本研究の目的は、アプリケーションの実用化事例を検討し高信頼マルチキャストの実用化における問題点を明確にし、プロトコルの開発改良を行なうための基盤を確立し実用化を推進することである。

1.3 本論文の構成

本論文は、全9 章から構成される。各章の内容は以下の通りである。

- 第2 章では、高信頼マルチキャストがもつ先天的な問題点について概要を述べる。
- 第3 章では、執筆時点でIETF を中心に行なわれている標準化の動向について説明する。
- 第4 章では、既に提案されている高信頼マルチキャストプロトコルの分類と代表的なプロトコルについて具体的に見ていく。
- 第5 章では、信頼性が求められるファイル転送アプリケーションの設計事例をあげ、具体的な問題点を指摘する。

- 第6 章では、実時間性が要求されるアプリケーションの実験例から明確になった問題点、プロトコル拡張要求を説明する。
- 第7 章では、本研究で提案するツールキットの設計と実装を述べる。
- 第8 章では、本研究の提案に対する今後の課題を述べる。
- 第9 章では、本研究のまとめを述べる。

第 2 章

高信頼マルチキャストとその問題

高信頼マルチキャストプロトコルは、アプリケーションの多彩な要求、マルチキャスト方式が持つ特有の技術課題から標準化が困難とされている。本章では、高信頼マルチキャストが持つ問題について説明する。

2.1 多彩な要求

高信頼マルチキャストアプリケーションは、実時間性および配送するデータの種別から表2.1のように分類できる。表の左上に位置するアプリケーションは実時間性が求められるマルチメディアアプリケーションで、配送の信頼性が確保できなくても低ジッタな配送が優先される。表の右側のアプリケーションは配送の信頼性を重視する。

2.1.1 実時間性が求められるアプリケーションの例

ビデオやオーディオストリームの配信といったアプリケーションは低遅延、低ジッタな配送と実時間性が要求される。マルチキャスト技術の適用が期待されているマルチメディアデータは、実時間への依存度を基準に次のように分類できる。

- 分離型メディア (時間独立メディア)
- 連続型メディア (時間従属メディア)

分離型メディアとは、テキストやグラフィックの集まりなどを指す。連続型メディアは、音声や動画像など時間依存性の強いメディアを指す。連続メディアを扱うアプリケーションは、ネットワーク通信の要求を基準として次の 2 つに分類できる。

表 2.1: マルチキャストアプリケーション

	実時間	非実時間
マルチメディア	• ビデオサーバ(1対多、多チャンネル) • ビデオ会議(多対多)	• レプリケーション – ビデオサーバ、Webサーバ • コンテンツの配送(写真集配信など)
データのみ	• ニュース配信(緊急災害など) • 株式状況配信 • ホワイトボード • 実時間ゲーム	• データ配送 – サーバ間、サーバとクライアント間 • データベース同期(nフェーイズcommit) • ソフトウエア同期(分散CVS)

- 厳密アプリケーション
- 適応アプリケーション

厳密アプリケーションは、最大ジッタ、最大遅延時間などネットワークに対して絶対的あるいは統計的な性能の保証を要求する。適応アプリケーションは一定の品質(QoS)は期待するが実際に提供される通信の品質の変化に応じてその動作を調整できる。具体的な適応アプリケーション例としては、遠隔教育のためのビデオ配信や英語会話教室などの小人数ビデオチャット多少の画像品質が劣化しても目的とする知識伝達が行なえれば良いとされるアプリケーションがあげられる。通信品質の劣化に応じて画像データのサイズや送出間隔を調節する方法が考えられる。

2.1.2 信頼性が求められるアプリケーションの例

信頼性が求められるアプリケーションの例としてイントラネットを用いたサーバ間のファイル同期などが考えられる。夜間に複数拠点のサーバのデータを完全に同期させる。データの配送の信頼性や配送状況の確認ができることが望まれる。イントラネット内部で

の利用に限定するとセッション広告の枠組の導入や送信者認証といった要求は比較的小さくなる。

実用例として玩具流通業者[1]の衛星ネットワークを用いたソフト更新情報の販売店舗への配布、自動車製造業[2]によるディーラへの在庫情報、ソフトウエアの配布などがあげられる。

2.1.3 途中参加が可能なアプリケーションの例

ネットワークを用いた協調共同作業を支援するツールにホワイトボードがある。参加者は、互いに共有しているホワイトボードへの描画動作を行なう事ができる。このアプリケーションではセッションの途中から共同作業に参加することができなくてはならない。途中参加した共同参加者には、それまでの描画内容が転送される。

また、現在、有線放送で行なっているような連続的な音声配信をマルチキャスト技術を用いて実現した場合でも、曲の切れ目まで待たされることなく途中参加が可能なことが望まれる。

2.2 フィードバックの爆発

フィードバックの爆発は、高信頼マルチキャストの設計を複雑にする代表的な要因の一つである。

マルチキャストのパケット配送は、送信者を根とする配送木を用いて行なわれる。送信者が送信したパケットは配送木を下流に向かって配送され受信者に到達する。

配送の信頼性を確保するために再送を行なう場合、受信者は肯定応答(ACK)もしくは否定応答(NAK)を送信者へ返送する。受信者から送信されたACKもしくはNAKパケットは配送木を上流に向かって送信者へ配送される。このとき送信者付近のネットワーク要素や送信者にパケットが集中するためパケットが欠落する。この様子を図2.1に示す。

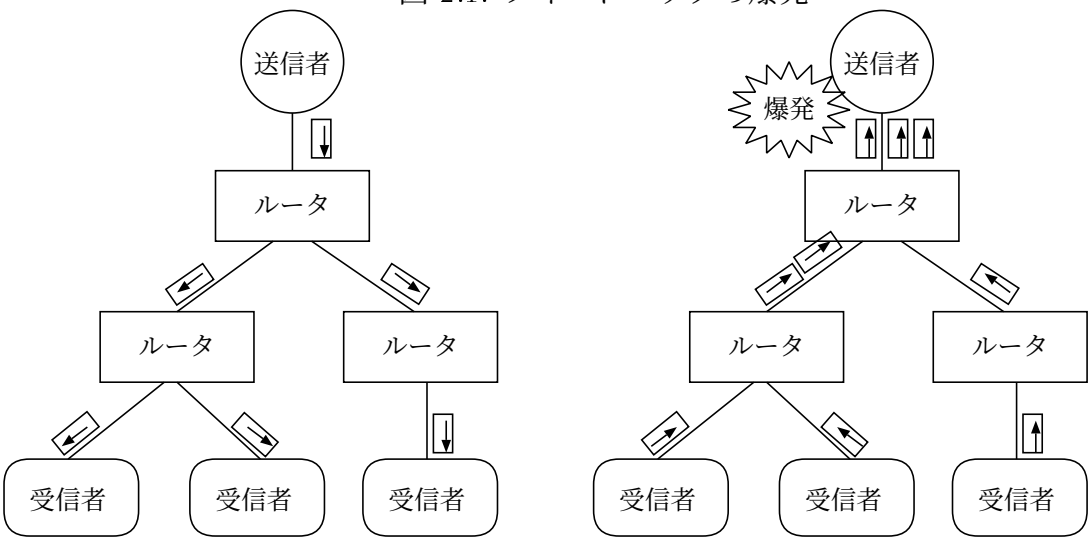
フィードバックの爆発を起こす要因としては、上で述べた肯定応答、否定応答の他に流量制御、輻輳制御を行なうための受信者から送信者へ送られる制御メッセージがあげられる。

また、送信者起動のプロトコルにおいては、送信者にフィードバックされる受信者の参加離脱メッセージもフィードバック爆発の原因となりうる。

[1] トイザらス社

[2] General Motors社

図 2.1: フィードバックの爆発



フィードバックの爆発は、受信者がACK、NAKパケットの送出時刻をランダムに遅延させる方法やルータでフィードバックパケットを抑制する方法で防止される。

高信頼マルチキャストプロトコルは、フィードバックの爆発を防止する方策を内包しなくてはならない。

2.3 TCPとの親和性

パケット単位で最善努力型の転送を行なうインターネットに展開する新たなプロトコルは、そのサービスの低下や停止(Internet meltdown)をひき起こすことがあってはならない[8]。本節では、TCPの輻輳制御、マルチキャストプロトコルで考えられる輻輳制御の手法について概観する。

2.3.1 TCPの輻輳制御

TCPにおける輻輳制御(輻輳は回避するものであるが)は、ACKベースの誤り再送制御機構とともにスライディングウインドウ方式で実現されている。[3]TCPの送信者は、受信者からのACKの不到達でパケットの損失および輻輳の発生を検知する。輻輳を検知した送信者は、投機的に送付するパケットの数を小さく抑えるスロースタートのアルゴリズムを駆動する。具体的には、TCPの送信者は、輻輳を検知すると輻輳ウインドウのサイズを1にして投機的パケット送信量を抑える。パケットの損失が検知されなければ受信者

から広告されたウインドウサイズまで送信量を徐々に増大させる。

輻輳リンクへの入口のルータが、輻輳発生時に流入する複数のTCP フローのパケットを出力キューで公平に欠落させる機構[15]を持っていれば、複数フロー間で公平な通信が行なわれる。

2.3.2 マルチキャストのフロー・輻輳制御の手法

マルチキャストのフロー・輻輳制御としては次の4つが上げられる[16]。

- フロー・輻輳制御を行なわない方式
  この場合、送信者は定レートでパケットを送付し続ける。送信者が送付するパケットの一部は輻輳リンクを通過する際に損失を起こす。送信者からみて輻輳リンクの向こう側に位置する受信者に対しては、無駄なパケットを送付していることになるが、輻輳リンクの手前に位置する受信者に対してはアプリケーションが要求する時間にパケットを到達させるために送付していることになる。

  定レートのフローがTCP フローと混在する場合、輻輳発生時には、TCP が送信レートを絞るため、定レートフローが帯域をほとんど占有してしまう。

- ネットワーク内の中継ノード(ルータ)が過剰なパケットを破棄する方式
  ルータが次リンク帯域容量を管理し、フロー毎に分配する。ルータでは優先キュー機構を用いてフロー毎にキューを割り当てるぞれぞれのキューから次リンクへの出力はフロー毎の設定レート以上は行なわないので、設定レート以上過度に流入するパケットは破棄される。

  この方式では、同じリンクを共有するTCP フローを不当に圧迫することは避けられる。

  この方式は、次節で述べる統合サービスアーキテクチャの取り組みで標準化されようとしている。

- 送信者にフロー制御情報・輻輳制御情報をフィードバックし、送信量を制御する方式
  フィードバックされた情報により輻輳を検知した送信者は、送信の一時停止/ 送信レートの低減を行なって輻輳を回避する。フィードバックパケットを送付する対象は、パケットの到着レートやキュー長などを監視している中継ノードと受信端末(エンド・エンドの制御)があげられる。

上に述べた2つの方式と比較して無駄なパケット送付の削減は期待できるが、次にあげる問題点がある。

- フィードバックの爆発が発生しうる。
  中継ノードでフィードバックを圧縮する機構がない場合を想定する。送信者からみて輻輳リンクの向こう側に位置する受信者は、一斉にパケットの損失を検知する。受信者から送信された制御パケットは送信者もしくはその付近のネットワーク要素で爆発をひき起こす。
- フィードバックの遅れによるオーバーシュート
  上でみたようにスライディングウインドウ方式のTCPの送信者は、輻輳を検知すると即座に輻輳ウインドウのサイズを最小(1)にして、送出量を抑える。この機構と親和性を持つマルチキャスト輻輳制御プロトコルを考案することは難しい。

  今、マルチキャストのエンド・エンドの輻輳制御を考える。マルチキャストに参加する受信者は、パケットの損失を機に、検知した輻輳を送信者へ制御パケットで知らせる必要がある。中継ノードでフィードバックを圧縮する機構がない場合、受信者はフィードバックの爆発を回避するためにバックオフ機構を採用する。バックオフ機構は確率的にフィードバックパケットの送付を遅らせる。フィードバックが遅れると送信者にフィードバックパケットが届くころには、他のTCPフローの輻輳制御により輻輳が回避されているかもしれない。

  次に中継ノードでフィードバックを圧縮する機構がある場合、中継者が転送するフィードバックパケットの遅延が問題となったり、抑止すべきパケットを同一視する方法が困難になる。中継者が下流に位置する受信者や中継者のすべて知っている場合、即座に輻輳検知のフィードバックパケットを送信者に送付することが可能である。この場合、輻輳を検知した最初の受信者からのフィードバックで輻輳制御が起動される。中継者が下流に位置する受信者や中継者を知らない場合、セッション中で検知した輻輳を同一視するための機構が必要となる。中継者は、この機構により同一視されたフィードバックパケットを圧縮することができる。ところが、パケットの損失を機としない輻輳を受信者で共通に検知することは難しい。受信者で輻輳を検知する際の問題点を次に述べる。
- エンド・エンドの制御の場合、受信者の能力不足と輻輳を別に検知することは困難である。
  TCPでは、受信者はパケット損失を重複ACKを発行する。送信者は重複ACK

を連続的に受信するかタイムアウトで損失を検知する。TCPでは、フロー制御、再送制御、輻輳制御が一体となって実現されているので、パケットの損失と輻輳を分けて検出する必要はない。

受信者がパケットの損失と輻輳を別に検知することは難しい。例えば、受信者の受信バッファの量を越えてバーストパケットが到着した場合、パケットは損失する。この損失が輻輳リンクを経由してきたパケット群によりひき起こされたのか、何らかの理由で受信者の処理能力が不足したため発生したのか切り分けることはできない。[3]パケット損失を元に輻輳を検知する場合、次の2つの方法が考えられる。

* 一つのパケット損失を輻輳の発生とみなす。
  この場合、上に述べた問題を内包する。
* パケット損失率で輻輳を検知する。
  この場合、上に述べた問題の影響を緩和することができるが、フィードバックの送付が遅延する。

輻輳の発生を送信者に知らせるフィードバックが遅れると送信者はオーバーシュート状態になる。

- マルチキャスト経路制御を用いた方法
  データストリームを複数のバンドに分割し、それぞれを異なったマルチキャストグループに送出する。受信者はすべてのグループに参加すればもとのデータがすべて受信できる。経路上に輻輳が発生すると輻輳リンクより下流の受信者が一斉に輻輳を検知して一部のマルチキャストグループから離脱する。このときマルチキャストの経路制御により輻輳リンクより下流には離脱したグループのパケットが流れなくなるため輻輳リンクを流れるトラフィックが減少し、輻輳制御として動作する。

  この方式では、マルチキャストセッションを構成している受信者の配置によって、輻輳回避の動作が遅延する。すなわち、マルチキャスト経路制御プロトコルは、端末が参加・離脱を要求してから経路情報が伝搬して輻輳リンクに伝わるまでは遅延が発生する。

今まで見てきたように、TCPと公平に帯域を共有するマルチキャスト輻輳制御プロトコルを考案し、標準化することは困難な作業である。[4]

[3]一定の能力幅の受信者群を仮定できない。

[4]IETFの高信頼マルチキャスト作業部会では輻輳制御の構築ブロックに関する草案について議論中である。

2.4 セキュリティ要求

一般に高信頼マルチキャストプロトコルに関する要求とセキュリティに関する要求は直交する概念なので独立に検討することができる。したがって、本研究で言及する範囲から外れるが、高信頼マルチキャストを利用したアプリケーションの要求とは密接に関連するので、ここで基本的な事項について整理しておく。

マルチキャストパケットの配送は、参加者が特定のマルチキャストグループに参加する要求を発行することで開始される。受信者から発信された参加要求のメッセージはルータで解釈され受信者が属する自立ドメイン内部で採用されているマルチキャスト経路制御プロトコルの方策にしたがって配送木が構成される。送信者はマルチキャストグループに参加している受信者のメンバ管理に関与しない。送信者が送付した相手先アドレス欄にマルチキャストグループアドレスを記述したパケットは、ルータによって解釈され配送木に添って転送される。受信者は、マルチキャストセッションの途中であっても動的に参加・離脱ができる。この枠組で実現されているマルチキャスト配送では次の2点が問題となる。

- 参加要求を出した受信者のメンバ管理されない。
- マルチキャストグループに宛てられたパケットはすべての受信者に配送される。

前者は、配送されたデータに対する課金を困難にしている。暗号キーを交換を拡張して多数の受信者に鍵を渡しマルチキャストを行なうホストグループの動的なメンバーシップ管理を行なう適切な方法は、まだ見つかっていない。課金の公平性から機密性を要するマルチキャスト通信では、送受間でのセキュリティ連携を確立してから、データ配送を開始する。セッション中に受信者が参加・離脱・再参加するためには、そのたびにセキュリティ連携を確立しなおす必要がある。[5]

後者は、例えば、ソフトウエア配送を行なうアプリケーションのセッションに悪意を持ってウイルスを混入させるといった危険性を示している。また、サービスの品質を低下させる目的で多量のパケットを特定マルチキャストグループに送付しすることが可能である事を意味する(DoS)。これに対しては多量のパケットを送付するコストが攻撃を試みる者があきらめるに程度十分大きくなるような対策が必要となる。具体的には、IPv6で提案されている認証ヘッダを用いる方法が考えられる。[6]

[5]送信者起動や集中型のメンバシップ管理を行なう一部の高信頼マルチキャストプロトコルでは、鍵配送を効率的に行なう事が可能である。

[6]認証ヘッダはRFC2402で定義されている

第 3 章

IETF を中心とした標準化の動向

Internet Engineering Task Force (IETF) では高信頼マルチキャスト作業部会を中心に高信頼マルチキャストプロトコルに関する標準化が進められている。代表的なマルチキャストに関連する作業部会としては、自律ドメインを越えたマルチキャスト経路制御を扱う BGMP 作業部会、MBONE の普及展開のための作業部会、受信者側から送信者に至る経路探索を可能にする MSDP の作業部会、マルチキャスト経路制御プロトコルである PIM を扱う作業部会、マルチキャストアドレス割り当てに関する作業部会などがあり、それぞれ草案の作成と標準化 (RFC 化) の作業を行なっている。ここでは、本研究に関連の深い取り上げてまとめる。

3.1 大規模マルチキャスト アプリケーションの通信要求の分類

マルチキャストアプリケーションの要求は複雑である。これがプロトコルの標準化を困難にしている。[10] では、大規模マルチキャストの通信の要求を定性的に定義するための指標を提示されている。アプリケーションの要求を正確に定義しクラス分けする事は標準化への第一歩である。この RFC で定義されている内容を表 3.1、表 3.2、表 3.3、表 3.4 に示す。

表 3.1: マルチキャストアプリケーションの要求定義1

要求	単位	意味
信頼性(パケット損失関連)		
トランザクション的	2値論理値	『関連する複数の操作がすべて完了するか、一連の操作のうち一つでも失敗した場合、すべての操作が行なわれず失敗した事が知らされる』ということが必要かどうか
保証された	列挙	『成功するまで繰り返し実行される』、『関連する要素が正常に動作していれば必ず成功することが保証される』、『何も保証しない』、のいずれが要求なのか
寛容な損失	割合	アプリケーションが使いものにならない状態にならないために許容される最大の損失率
意味的損失	識別子	アプリケーションのデータのどの部分がどのぐらい無視されうるかを示す
信頼性(構成要素関連)		
セットアップ失敗	時間(秒)	回線接続、JOIN処理など通信開始までに許容される時間(例外処理が起動されるまでの時間)
平均故障間隔	時間	その通信チャネルが故障して復旧するまでの平均時間
ストリームの失敗時間	時間	回線切断と判断するタイムアウト
順序保証		
順序保証の型	タイミング	タイムスタンプで保証(包括的に、送信元毎で)
	シーケンス	生起順序で保証(包括的に、送信元毎で)
	因果	因果として保証(包括的に、送信元毎で)

3.2 高信頼マルチキャストトランスポートの標準化の動向

高信頼マルチキャストに対する要求は多彩である。IETFの高信頼マルチキャスト分科会では、プロトコルの評価指標に関する標準化を終了し、1999年3月の会議で具体的なプロコトルの標準化に関する方針が話し合われた。その中で、すべての要求を満たすたwhole protocol方式ではなく、必要な要求を満たす構築ブロックを個別に開発標準化する方式が採択された。[9]で高信頼マルチキャストの評価指標についてまとめられている。この中でwhole protcol方式採用すべきではないという結論が述べられている。表3.5に、このRFCで述べられている高信頼マルチキャストが満たすべき性質についてまとめたものを示す。

次に高信頼マルチキャスト分科会が提出している草案(Internet Draft)について概観する。

3.2.1 構築ブロックに関する草案

バルクデータ転送に関する高信頼マルチキャスト転送方式に関する構築ブロックについて、他のプロトコルで共通に利用できる構築ブロックの特徴を抽出することを目標にまとめられている。

草案の中では、構築ブロック方式の利点と欠点、以下に示すようなプロトコルの4つのクラスへの分類、構築ブロックの候補について説明されている。

- ネットワークの援助なし
  マルチキャストセッションを構成する送信者、受信者間のプロトコルだけで損失を回復する。受信者の数に対するスケーラビリティを確保するため、損失回復に必要なトラフィックを削減する機構を含む。SRM,MDP2などが代表的なプロトコルである。

- サーバ援助
  肯定応答を用いて、高い配送の信頼性を提供する。スケーラビリティの確保のためには、サーバを階層状に配置する必要がある。RMTP,RMTP-II,TRAMなどが代表的なプロトコルである。

- ルータ援助
  否定応答を用いてパケットの配送の信頼性を向上させる。否定応答の損失が繰り返される可能性があるので送信者から見た完全な信頼性を提供することは出来ない。この方法の利点は、新たに開発されるルータのソフトウエアは否定応答(の圧縮)と再送パケット(による状態の開放)のみ扱えば良いという点があげられる。代表的なプロトコルとしてはPGMがあげられる。

- オープンループ配送
  前方誤り訂正(FEC:Forward Error Correction)に代表されるように、受信者からのフィードバックなしに、送信データの信頼性を向上させようとするプロトコル。このような手法にもとづいたプロトコルは衛星回線のような非対称ネットワークとの親和性が高い。

3.2.2 設計空間に関する草案

多彩な高信頼マルチキャストアプリケーションの要求がある中で、特定の要求に着目してプロトコル、システムを設計実装することは、広大な設計空間の中から要求に応じた特別なベクトルを規定することに対応する。

この草案では、バルクデータ転送に関する構築ブロックの標準化の足掛かりとして、プロトコルの設計空間を大きく規定する指標、要求が記述されている。規定される設計空間については、前節で述べたプロトコルクラスの類型毎に解説されている。

記述されている指標の一覧を表3.6、表3.7に示す。

3.2.3 ルータ支援に関する草案

一部のエンド・エンドのマルチキャストプロトコルでは、ルータ支援を付加するとトラフィックの圧縮が出来る。この草案では、ルータ支援のプロトコルクラスが与える解空間の範囲を明確に定義し、具体的なルータ支援機能の定義とアーキテクチャについて解説している。このクラスのプロトコルの実例としてPGM(Pragmatic General multicast)があげられる。これについて具体的に次章で説明する。

3.2.4 今後提出される予定の草案

高信頼マルチキャスト作業部会は2000年2月に次回会議の開催を計画している。次回、会議以後に次のような草案が提出される予定である。

- 肯定応答ベースのプロトコルクラスに関する草案
- 否定応答ベースのプロトコルクラスに関する草案
- オープンループ配送のプロトコルクラスに関する草案
- 構築ブロックのプロトコルインスタンス方式に関する草案
- 階層マルチキャストでのデータ転送を可能にする符合化(ALC)に関する草案
- 輻輳制御に関する草案

3.3　統合サービスアーキテクチャで提供される通信の品質

IETF では最善努力型の配信以外のサービス提供できるという確信のもと統合サービスアーキテクチャ、区分化サービスのアーキテクチャを発表している。通信パス上のルータが統合サービスアーキテクチャに準拠した場合、次節でしめすサービスクラスが適用される。特定フローに対して上記サービスを割り当てる枠組が区分化サービスのアーキテクチャとして、標準化のため提案されている。[1]

3.3.1　サービスのクラス

統合サービスワーキンググループが定義しているサービスクラスの代表的なものを表3.8に示す。提供されるサービスの質によってアプリケーションが実装すべき機構に違いが出る事がわかる。

[1]帯域幅の先物取引市場が形成されるかもしれない。利益目的で投機的行動を行なう者が現れれば、同時にリスクヘッジ商品も企画される。もしくは、航空機の格安チケットのような市場になるかもしれない。それとも、流行りの逆オクション形態?

表 3.2: マルチキャストアプリケーションの要求定義2

要求	単位	意味
実時間性		
厳しい実時間性	2値論理値	通信が開始される前に実現可能かどうかが報告される。弱い実時間性は確率で規定される。
同時性	時間	複数のストリーム間のずれ
バースト	割合	変動する帯域幅を割合で示す
ジッタ	時間	許容される最大変動時間量を示す
消滅	日付	情報が有効である期間を示す
レイテンシ	時間	アプリケーションからみた初期化時刻と実行時刻の差異
理想的な帯域幅	帯域幅	通信を完了するのに必要な帯域幅
肝要な帯域幅	帯域幅	アプリケーションが動作可能な最低帯域幅
要求される時刻と寛容性	日付	通信が完了してほしい時間。通信が完了すべき時間
ホスト性能	アプリ	通信を創造/消費(送信/受信)するためのホストの能力をアプリケーションの水準値で示す
フレームサイズ	データサイズ	アプリケーションからみた論理的なデータパケットのサイズ
コンテンツサイズ	データサイズ	通信する対象物のサイズ(例:ドキュメントの大きさ)
セッション管理		
初期化	時間	広告、勧誘、参加指示を行なう時間を列挙する
開始時間	日付	送信者がデータを送付するを開始する時間
終了時間	日付	送信者がデータを送付するを終了する時間
継続期間	時間	終了時間-開始時間
活動時間	時間	セッション中の中断時間を除いた時間
セッションバースト	割合	セッション中に期待されるバーストレベル
原始的な参加(率)	割合	セッションを開始するために最低限必要な参加者を参加者の候補と実際に参加者している参加者の割合で示す
途中参加許容	2値論理値	途中参加を許容するかどうか
一時離脱許容	2値論理値	一時離脱を許容するかどうか
途中追い付き参加許容	2値論理値	追い付き更新を伴う途中参加を許容するかどうか
先天ストリーム数	整数	セッション中で発生しうるストリームの本数
活性ストリーム数	整数	セッション中で同時に活動するストリームの最大本数

表 3.3: マルチキャストアプリケーションの要求定義3

要求	単位	意味
セッショントポロジー		
送信者数	整数	ミドルウエアが許容する最大の送信者数
受信者数	整数	ミドルウエアが許容する最大の受信者数
ディレクトリ		
失敗タイムアウト	時間	信頼性と同様
流動性	列挙	ディレクトリ要素ががいつ書き変わってもよいかの制限を定義する
セキュリティ		
認証の強さ	抽象貨幣	ある役割を乗っ取るために必要なコスト
いたずら補強	抽象貨幣	データを改定破壊するためのコスト、データを再生するために必要なコスト、実時間性を阻害するために必要なコスト
拒絶しない強さ	抽象貨幣	時刻、順序、宛先、内容などについて拒絶するために必要なコスト
サービス妨害	抽象貨幣	サービス妨害をするためのコスト
活動の制限	メンバリスト	その役割をそのメンバが行なう事ができるかどうか
プライバシ	抽象貨幣	誰かわかる、誰かはわからないが同じ人とわかる、同じ送信元とわかる、検知できない、と言う状況をつくり出すために必要なコスト
極秘性	抽象貨幣	通信の秘密を暴くために必要なコスト
再送予防の強さ	抽象貨幣	あるデータを再送させるために必要なコスト
参加者管理基準	マクロで	ユーザのリスト、組織に属するユーザのリスト、ホストのリスト、ユーザの性質を規定する規則、組織を規定する規則、ホストを規定する規則で表現する
共謀防止	抽象貨幣	時間競合、暗号鍵共有時の競合、QoS確保時の競合を発生させるためのコスト
公平性	様々	参加者が途中参加できること、リアルタイムゲームですべての参加者に同じ遅延で情報が伝わること、など様々な規定がある
危機時の行動	列挙	危機を検出したときにとる動作

表 3.4: マルチキャストアプリケーションの要求定義4

要求	単位	意味
セキュリティ力学		
平均危機時間	時間	そのシステムが危機状態になる平均時間。(クラックしたくなるようなシステムでは数値は高くなる)
危機検出時間制限	時間	システムが危機状態になったことを検知するのに必要な時間の平均
危機回復時間制限	時間	危機状態になったシステムが再び安全な状態になるまでに必要な時間の上限
支払と課金		
全コスト	通貨	通信でかけることのできるコストの上限
時間ごとのコスト	通貨/時間	単位時間あたり課金されうるコストの上限
Mb ごとのコスト	通 貨/デ ータサイズ	通信で課金されうるデータサイズ毎のコストの上限

表 3.5: 高信頼マルチキャストプロトコルへが満たすべき性質

性質	条件
輻輳制御	インターネット上で安全に展開するために特に次の3つをみたさなくてはならない。a) 良いスループットを得られること。(不謹慎なデータ転送や再送トラフィックでリンクに過負荷をかけないこと) b) リンクの利用効率を高めること c) 競合するフローが『餓死』しないこと。
スケーラビリティ	プロトコルは複数のネットワーク技術、リンク速度、受信者の数を含む様々な条件の元で機能しなければならない。どうやって、いつプロトコルが働くなる条件を理解する事は重要である。
安全性	プロトコルの分析のなかで安全性、プライバシの保護の方法を示さなくてはならない。機密保護、否定サービス攻撃対策についても提供される必要がある。
順序保証	プロトコルは送信元ごとの順序保証するかどうかを示さなくてはならない。複数送信者の総合的な順序づけはサポートすることは推奨されない。それはプロトコルをスケールすることを難しくする。上位層で実現されるほうがより簡単である。
ネットワークトポロジ	プロトコルはインターネット全体に適用されても動作する必要がある。初期の適用はイントラネットが想定される。それゆえ衛星ネットワークのサポート (地上網で返信パスがあるばあいと返信パスがない場合) が推奨されるが、要求はされない。
グループメンバー管理	グループのメンバー管理のアルゴリズムはスケーラブルでなくてはならない。メンバー管理は匿名を許す (送信者は受信者をしらない) もしくは完全配送 (送信者は受信者の数を受信して、付加的に失敗者も把握される。)
アプリケーション例	マルチメディア放送、実時間経済市場データ配送、複数ファイル転送、サーバー複製といったサンプルアプリケーション

表 3.6: 設計空間を規定する要求

要求	意味
アプリケーション要求	
すべての受信者が受けとったか?	受信状況を確実に把握する必要があるかどうかという要求を示す。詳細に受信状況を把握する必要がなければ、肯定応答の集約が設計指針として採用できる。アプリケーションによってはパケット単位ではなく意味のあるデータ単位の受信状況だけが必要なのかもしれない。
配送状況の差異が問題になるか?	株式市況の配信などでは受信者間に伝わった情報の差異が問題になるかもしれない。実現するためには受信者が統計を取得し、それらを比較する枠組が必要となる。
受信者の数に対するスケーラビリティは必要か?	想定する受信者の数が少なければ、フィードバックの爆発を懸念して肯定応答、否定応答のバックオフ機構を導入する必要がないかもしれない。階層的なフィードバック圧縮を行なうプロトコルの階層の深さは受信者の数で規定できる。前方誤り訂正(FEC)は、様々な受信者がそれぞれ異なったパケットを損失したときに効果を最大限発揮する可能性を持っている。
完全に高信頼か、一部高信頼か?	データ転送のようなアプリケーションでは一部のデータ欠けも許されない。しかし、音声、画像データの配送では、多少の画像欠けは許容される。

表 3.7: 設計空間を規定する要求

要求	意味
ネットワーク制約	
インターネットかイントラネットか?	イントラネットではずべてのネットワーク機器の管理権限がある状況を想定できる。インターネットでの展開を計画するときは一度にすべてのルータを置き換える方策は選べない。
帰り道	受信者から送信者への経路が送信者から受信者への経路と同じであると仮定できるかどうか。ネットワークの中間ノードが送信者下流に向けたパケットと逆向きのパケットで状態を変更するようなプロトコルは、非対称な経路に弱い。経路制御プロトコルによっては、この仮定は成立しないこともある。トンネル技術の利用や衛星回線のような非対称経路の存在は設計空間を規定する厳しい要求の一つである。
ネットワーク要素の支援を必要としているか?	階層化した複数のマルチキャストグループを使うか、サーバ主体の手法か、ルータ主体の手法か

表 3.8: 代表的なサービスクラス

サービス	意味
最善努力型	サービスの品質は保証されない。ネットワーク負荷の増大するにつれ、遅延の増加、パケット損失が発生する。
制御負荷サービス	負荷の軽いネットワーク上で最善努力型フローによって提供されるサービスと同等のサービスを提供する事を保証する。適応型のリアルタイムアプリケーションに向く。制御負荷サービスを必要とするフローでは、トラフィックの特性を記述したTSpecをルータに受け渡す必要がある。ただしTSpecにはピーク伝送率パラメータを記述する必要はない。 このサービスは、ISPが顧客に対して提供する事を前提に考案されている。サービスを実現するためには、ISPの境界ルータ契約した顧客のフローパケットに印づけを行なう機構、ISPの内部ルータで印づけされたパケットを確率的に落ちにくくする機構を用いて比較的容易に実現できる。
保証サービス	フローのトラフィックが指定したトラフィックパラメータの範囲内にあるかぎり、保証された配信時間内にパケットが到着し、ルータキューの溢れを原因とするパケットの損失がないことを保証する。最小、平均遅延の制御を行なうことや、ジッタの最小化を行なう事を保証するのではなく、最大遅延を保証する。このサービスを利用すれば、固定化されたプレイアウトバッファを持つ実時間アプリケーションの動作を保証できる。(アプリケーション自身の動作は別に保証されていると仮定する。)

第 4 章

既存の高信頼マルチキャストプロトコル

本章では、代表的な高信頼マルチキャストプロトコルの特徴的な動作について概観する。

4.1 既存の高信頼マルチキャストについて

4.1.1 Application Level Framing(ALF)の必要性

マルチキャストトランスポートに関する代表的な見解であるアプリケーション層のフレーム化(Application Level Framing)[20]について述べる。

ALFはアプリケーションとトランスポートとの間を厳密に層に分けるのではなくより協調的な動作を行なわせるべきという主張である。

ユニキャストでは、TCPがすべての要求を満たす一つのプロトコルとして使われている。ユニキャスト通信では、送信者、受信者が対になって通信を行なっている。したがって相手との通信ができなくなった時点で通信の継続不能状態を検知できる。しかしマルチキャストでは、多数の様々な受信者が、送信者が送出するパケットを受信している。ある受信者が受信不能状態に陥ったとして、送信者や他の受信者がそれを検知すべきかどうか、検知した場合に、セッションを中断すべきかどうかなどは、アプリケーションの要求に依存する。また、TCPでは送信者、受信者の間で番号づけを行なったパケットを配送し、パケット単位での損失回復、重複回避、順序保証が行なわれている。マルチキャストアプリケーションの中には、受信者がセッションの途中からの参加できることを要求するものがある。この場合、参加時点で受信したパケットに付与されている通番がアプリケーションの中でどんな意味を持つのか確定できない。(例えば、大量のファイル転送を行なっている場合など、通番が巡回してしまうため、ある通番を見ただけではセッション中のど

の位置のデータであるかが特定できない。) したがって、マルチキャストアプリケーションの多彩な要求に答えるためには、アプリケーションは能動的に、トランスポートに関する制御を行なうべきであるし、トランスポート層は、アプリケーションの要求するデータ単位 (ADU:Application Data Unit) の配送行なうべきであるという主張が ALF である。

このアプローチは特に実時間アプリケーションの処理に関しては有効であるとされている。

4.1.2 損失回復の方法による分類

高信頼マルチキャストプロトコルは多数提案されている。これらを特徴づけし分類する方法としては、送信者ベース (肯定応答ベース) か受信者ベース (否定応答ベース) かという方法や、受信者を配置するトポロジが木構造か、雲状か、リング状かという方法がある。送信者、受信者が参加するマルチキャストグループの直径が大きくなった時には、損失を発生させた受信者に近い受信者が再送を行なう方式 (ローカルリカバリ) が全体のトラフィック削減に有効である。ローカルリカバリに特徴を持つ高信頼マルチキャストプロトコルも存在する。

ここでは、損失回復の特徴に着目して次のように分類した。

- 肯定応答ベースで再送を行なうプロトコル
  送信者は、すべての受信者から肯定応答を受けとりその状態を管理する。パケットの損失は受信者から送信者に報告されて、送信者は再送を行なう。受信者の数に対するスケーラビリティを確保するためには、受信者を階層的に木構造に位置してフィードバックの爆発を防ぐ。中継者に受信者への配送責任を委任し ACK の集約を行なうプロトコルは、受信者の障害やネットワークの分割に対してうまく対応できる。肯定応答ベースのプロトコルは受信者の数が増加するにつれて、送信者が管理すべき状態数や肯定応答を処理するための帯域などが増加するので最大スループットは劣化する。

- 否定応答ベースで再送を行なうプロトコル
  送信者は、受信者の状態を管理しない。受信者はパケットの損失を検知すると否定応答を発行してパケットの再送を促す。配送の信頼性を高めるために送信者がデータをメモリ (送信ウインドウ) から消去するまえに、ボーリングを行なうプロトコルもある。受信者の数に対するスケーラビリティは優れるが、受信者アプリケーションでは、配送が完全に行なわれない場合を想定した例外処理を行なう必要があるかもしれない。(アプリケーション要求による)

- 再送を行なわないプロトコル
  前方誤り訂正(FEC)のように予め冗長パケットを送付することで信頼性を高めるプロトコル。肯定応答、否定応答を送信者へ返送する必要がないので、非対称経路のネットワークや再送によるトラフィック増加や再送パケットが到着するまでの時間が問題となる場合には有効なプロトコルと言える。

- ローカルリカバリに特徴を持つプロトコル
  IPマルチキャストの配送範囲を指定する機能[1]を用いて、再送要求の送付範囲を限定する事ができる。損失を検知した受信者はまず小さい配送範囲で再送要求をマルチキャストする。該当するパケットの受信に成功している受信者は再送要求に答えて該当するパケットを再送する。小さい範囲で返答する受信者がない場合は、配送する範囲を広げて再送要求を繰り返す。

4.2 既存の高信頼マルチキャストトランスポートの特徴

ここで既に提案されている代表的な高信頼マルチキャストの動作、特徴を見ていく。

4.2.1 RMTPII

Reliable Multicast Transport Protocol II (RMTPII)[14]は小数から多数への信頼性の良いデータ配送を行なう肯定応答ベースのマルチキャストプロトコルである。受信者をregionという単位にグループ分けし階層化することにより木構造を形成する。各ノードにはノードを代表する制御ノードが割り当てられ、そのノードの受信者のメンバ管理、肯定応答処理、再送処理の責任を持つ。RMTPIIはパケットの到着の信頼性を高く保証する。また、確実にパケットを受信した受信者の数を送信者に報告する機構を持つ。RMTPIIでは損失率やRTT(Round Trip Time)などの情報を用いてTCPの輻輳制御と互換性がある輻輳制御の機構を実現している。

4.2.2 PGM

PGM(Pragmatic General Multicast)[17]は、Cisco System社が1998年のIRTFで発表した高信頼マルチキャストトランスポートプロトコルである。

[1]IPv4ではttl欄、IPv6ではホップ制限欄、スコープ欄を活用する

表 4.1: PGM のパケット型

パケット型	意味
ODATA	Original Data。データを配送する
RDATA	Repair Data。再送データを配送する
SPM	Source Path Message。PGM 配送木を決定する。
NAK	Negative acknowledgement。否定応答。
NCF	Negative acknowledgement confirmation。否定応答受領通知

PGM は否定応答を用いた再送制御により損失回復を行なうプロトコルで、到着順序が保証のされた、重複のない配送を行なう事ができる。PGM ではすべての受信者は送信元が送信するすべてのパケットを受信できるか受信できなかったことがわかることが保証されている。PGM はエンド・エンドで再送制御を行なうプロトコルで、ルータ支援を受けて、受信者の数に対するスケーラビリティを確保する。

PGM では表 4.1に示す 5 つのパケット型が定義されている。

送信者は定期的に SPM を送出する。SPM を受けた PGM を解釈するネットワーク要素 (ルータ) は自らのアドレスを SPM に追加してマルチキャスト配送木の下流に SPM を転送する。受信者 (あるいは PGM を解釈する下流のルータ) は SPM に記載されているアドレスを参照して、自らの上流の PGM を解釈するネットワーク要素を認識する。送信者は、配送すべきデータを ODATA に載せてマルチキャストする。ODATA には通番が付与されている。受信者はこの通番の隔たりで損失を検知し自らの上流の PGM を解釈するネットワーク要素に対して NAK をユニキャストする。このとき付加的に TTL を 1 にして NAK をマルチキャストすることもできる。NAK を受けとった上流の PGM を解釈するネットワーク要素 (ルータもしくは送信者) は NCF を送出して NAK が別の受信者から送付されてくることを抑止する。NAK の送出は、バックオフしてから行なわれる。NAK 送出をスケジュールしている受信者が NCF もしくは同じ NAK を受けとると NAK の送出を一時取り止める。(バックオフする) 送信者が NAK を受けとると受信ウインドウに保持しているデータを RDATA に載せて再送する。

NAK,NCF,RDATA が損失した場合、受信者は NAK をバックオフしているのでタイムアウトを契機に NAK を再発行する。

RDATA が受信者に届けば、受信者はバックオフしている NAK の送出を取り止める。予め設定しておいた閾値の回数 NAK を再発行しても RDATA が得られない場合、受信者は回復不能エラーをアプリケーションに通知する。

PGM では、受信者はパケットを受信できるか、受信できなかったことを検知でするか

図 4.1: Erasure Codeの生成式

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} g_{11} & g_{12} & \cdots & g_{1k} \\ g_{21} & g_{22} & \cdots & g_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ g_{n1} & g_{n2} & \cdots & g_{nk} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \end{pmatrix}$$

いずれかの状態になることが保証されている。

PGMを解釈するルータは、NAKを受信すると修正状態を内部に保持して同じNAK を2度以上受信しても上流のPGMを解釈するネットワーク要素へ送付しない。この内部状態はRDATAを受信したときに解除される。ルータはNAKを圧縮しトラフィックを削減するがパケットの再送に直接関与しているわけではないので多量のメモリを消費することはない。

4.2.3 FEC

前方誤り訂正(FEC:Forward Error Correction)は、ネットワーク転送中に予期されるパケット損失する補完するための冗長パケットを送信し、受信側で、元のコードを再構成することで、高信頼なデータ転送を実現する技術である。FECは、遠距離伝送で使われている誤り訂正符合化であるある Erasure Codeを用いて実現する事ができる。

Erasure codeについて説明する。(n, k)ブロックErasure Codeは、k個のソースパケット群から、その一部のk個を取り出す元のk個のソースパケットを復元できるようなn個の冗長パケット群を生成する符合化ブロックを示す。この符合化は、符合化したパケット群にもとのソースパケットが含まれる時に系統的(systematic)という。パケットの損失率が低い時に系統的なコード化を行なったものの復号化の平均コストは系統的でない符合化を採用した場合と比較して低くなる。(もとのパケットの復号化計算コストは0である)また、符合化されるブロックが次のような行列を用いた線形変換で算出されるときに、線形(linear)であるという。$\underline{x}$ を要素数kの行ベクトル、 $\underline{y}$ を要素数nの行ベクトル、Gを$n * k$の行列として、

$$\underline{y} = G\underline{x}$$

すなわち図4.1に示す式で表現され、Gの任意のk行の集合が完全な階数を持ち、任意の$\underline{y}$のk個の要素から$\underline{x}$が再生できるとき線形であるという。

Erasure codeは例えば字式で示されるVandermode Matrixを用いて生成できる。

$$g_{ij} = x_i^{j-1}$$

ただし、x_i は GF(p^r) [2]の要素。

FECでは、このような符合化を用いて作成された冗長パケットを送付するので損失が発生しても元のデータを受信側で再送無しに再生できる。(n, k) ブロックで符合化されたパケットから元のデータを再生するためには任意のk個のパケットが正常に受信できていればよい。したがって次のような利点を持つ。

- 受信端末によって受信できたパケットにばらつきがある場合でも冗長の範囲内であればパケットの再送なしにデータを再現できる。
- 冗長の範囲を越えて損失が発生した場合、再送の仕組みを組み合わせて信頼性を向上させる方法が考えられる。このとき冗長化符合を再送すれば、受信側で受信できたパケットにばらつきがない場合でもばらつきがある場合でもデータを再現できるので効率的である。

欠点として次のようことがあげられる。

- バースト損失に弱い。すなわち (n, k) ブロックで符合化した場合、n-k 個以上の損失には対応できない。
- 計算時間がかかる。符合化復号化の演算時間はkを増大させると比例して大きくなる。[3]

4.2.4 SRM

SRMは、アプリケーションにとって意味のあるデータ単位で信頼性の確保をおこなうべきであるというALF(Application Level Framing)の概念に基づき設計されている[18]。この考え方に基づき設計されたSNAP(Scalabe Naming and Announcement Protocol)は、配送されるデータの単位に階層的な名前づけを行い、その状態を広告するプロトコルである。SRMは、このSNAPで定義される名前空間の状態を広告し、アプリケーションが名

[2]GF() はガロア体(有限体)を表す

[3][19]では、代表的な計算機システムの計算時間の最大値を紹介している。これによるとPentium 133MHzを搭載した計算機システムでFreeBSD上の実装の復号化速度は、$\frac{9.573}{k}$MB/sである。

前空間に対応づけられたデータを共有するためのプロトコルである。SRMでは、データの損失が検知されたときに、再送を行なう事で信頼性を確保する。SRMで信頼性を確保する単位は、ADU(Application Data Unit)と呼ばれている。ADUにはシーケンス番号が付与されている。損失はこのシーケンス番号の隔たりで検知する。論理的に階層的に配置されたADUを格納する器としてコンテナが定義されている。コンテナにはCIDが連続的に付与されており、この隔たりからコンテナの損失を検知することができる。しかし、この方法だけではコンテナ中の最後のADUの損失を検出することができない。SRMでは、各コンテナ毎の状態をセッションメッセージとして広告し、しっぽの損失を検知する。

損失検知がされたら、再送を要求する制御パケットを送付する。再送要求はランダムな時間が経過した後にrepare requestと呼ばれる制御パケットをマルチキャストする。待ち時間は、セッションメッセージのタイムスタンプにより求められたノード間の距離($d_{s,a}$)を元にして、次式で示される範囲の一様乱数として計算される。

$$\lfloor C_1 d_{s,a}, (C_1 + C_2) d_{s,a} \rfloor$$

ここでC_1, C_2は定数である。このタイマが時間切れになる前に他のノードが先に時間切れを起こし同じrepare requestがマルチキャストされた場合には、repare requestは指数バックオフされ制御パケット送付が抑止される。要求されたデータを持つノードは、同様に距離にもとづいて計算されるランダムな時間待ってrepare packetを送付する。その範囲は上図のC_1, C_2をD_1, D_2で置き換えたもので表すことができる。[4]

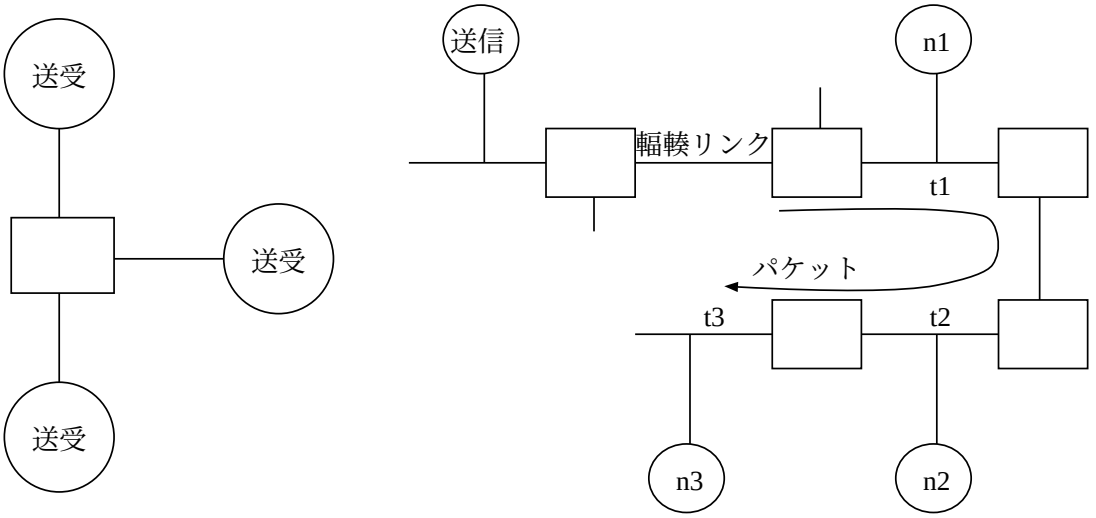
ネットワークトポロジーとSRMの再送動作について説明する。SRMはパケット損失要求を行なったノードに一番近いノードが高い確率で応答する機構をもつ。SRMの特性を明らかにするために、上で定義した、C_1, C_2, D_1, D_2およびネットワークトポロジの関係について解説する。

まず、ネットワークトポロジーとして均一な状態、すなわち、すべてのノードがスター状のトポロジーで接続されている状態では、すべてのノード間の距離が等い。したがって、同じ意味のパケットがネットワーク上に送出される機会を減らすためにはC_2, D_2を大きくすればよい。

次に、図4.2のようにノードが一列の鎖状に配置された例を考える。図では、あるリンクが輻輳しており、あるパケットが損失し、輻輳リンクを越えてシーケンシャル番号が飛んだデータパケットが到着している状態を表している。輻輳リンクに最も近いノード(n1)は、時刻t1に損失を検知する。次に接続されたノード(n2)はちょうどは、n1とn2の

[4] ノード間の距離の測定にはNTPで用いられている方法と同様の方法が用いられているので、ノード間で時刻が同期している必要はない。ただし、ノード間で転送されるパケットの経路が対称であることが必要である。

図 4.2: SRM のローカルリカバリとネットワークトポロジ



距離だけ遅れた時刻 t2 に損失を検知する。同様に次のノード (n3) は時刻 t3 に損失を検知するとする。C_1, C_2 がともに 0 の場合を想定すると、n1,n2,n3 がそれぞれ t1,t2,t3 に repare request を行ないネットワーク上に複数の repare request が送出される。C_2 を大きくする事で n2,n3 の repair request の重複を確率的に減らせる事がわかる。

最後にノード間の距離が不均一な例を考える。この場合 C_1(同様に D_1) を大きくする事でこれを抑止する事が期待できる。

SRM では、損失回復アルゴリズムの過去の振舞に応じてこれらの C_1, C_2, D_1, D_2 を動的に調整する適応アルゴリズムを提案している。

SRM は、広域ネットワークの不要なトラフィックを削減するローカルリカバリに特徴を持つプロトコルである。否定応答ベースのプロトコルなので損失回復に関して受信者が責任を負う。送信者はパケットが到達したかどうか確認しない。SRM では、マルチキャストの配送範囲の制限機能を利用したローカルリカバリの機構を有する。損失を検知したノードは、まず local scope と呼ばれる範囲に repare request を発行する。バックオフしているタイムアウト以内に repare data が到着しない場合は、グループ全体に repare request をマルチキャストする。

SRM は、名前空間を共有するアプリケーションを広域に展開するのに適したプロトコルといえる。

4.3 既存の高信頼マルチキャスト関連の技術

4.3.1 RLC

RLC: Reciever-driven Layered Congestion control は複数の IP マルチキャストグループを用いて、TCP フレンド リーな輻輳制御を行なうデータ転送方式もしくはそのモジュールの名称である。階層別に構成されたデータを複数のマルチキャストグループを使って異なった送信レートで送出する。受信者は、入手可能な帯域に合わせて、自律的に参加すべきマルチキャストグループを決定する。輻輳リンクの向こう側にあるすべての受信者にこの技術を適用すれば (同じプロトコルインスタンスが適用されていれば)、受信者が同期して行動し、輻輳制御が実現される。
送信者は、送信対象データを 0 から $n-1$ 層までの n 層のに分けて送信する。受信者は自らが参加する層 i を決定して 0 から $i-1$ 層までのマルチキャストグループに参加する。[5]期間 T ごとに、各層とも $B_0 2^i$ の速度でパケットを送出するノーマル期間、$2B_0 2^i$ の速度でパケットを送出するバースト期間、パケットを送出しない緩和期間を繰り返す。この T は、動的な輻輳制御を行なうために $T \leq 125ms$ を満たす必要があるとされている [21]。

4.3.2 ALC

ALC(Asynchronous Layered Coding) は RLC を用いて効率的にファイル転送を行なうための符合化技術である。2000 年 2 月 10 日の IETF 高信頼マルチキャスト作業部会の会合で取り上げられる予定である。詳細の情報は執筆時点では入手できていない。

[5]現在の入手可能な実装 http://www.iet.unipi.it/ luigi/mgpm/rlc990112.tgz においては、i層の送信速度は 0 層の送信速度を B_0 として、$B_0 2^i$ としている。

第 5 章

ファイル転送アプリケーションの実用化の検討

高信頼マルチキャストプロトコルの実用化の問題点を明確にするため、論理的に受信者を階層状に配置し、連続してファイル転送を行なうアプリケーションを設計、一部実装した。本章では、その要求定義と顕在化した問題点について記述する。

5.1 想定した要求

現在、紙媒体で行なわれている新聞の配送をインターネットで行なう事を想定した。1日の新聞のデータ量は2MB〜3MB と言われている。これを夜間の内に数百万世帯に配送する。そのような状況が実現できれば、同程度のデータ量を持った広告が新聞配送に続いて行なわれるようになるかもしれない。したがって、アプリケーションは数MB 程度のファイルを連続して送付する機能が必要とされていると仮定する。課金のため配送状況は送信元に伝えられる必要がある。

5.1.1 TCP との親和性

現在、インターネットは複数のISP により運営されている。配送のサービスを開始する企業からみて、サービスを受ける企業、団体、個人顧客は、複数の自律ドメインの向こう側に存在する。現在のインターネットでは、TCP フローを中心としたトラフィックを努力最善型のサービスで配送している。ここで考えるファイル転送機構は、TCP との公平な帯域共用を行なう必要がある。

5.1.2 大量ファイルの転送

議論:大量ファイルの同報は必要か

大量ファイルをマルチキャストする要求について考察する。夜間にサーバを同期させたいという要求は耳にする。製造業の分散開発を支援するため技術部門の複数CAD サーバを同期させたり、商品カタログを作成するための素材情報を製造元と流通で共用するなどの要求があげられる。しかし、ここに上げた要求例は、双方ともコンテンツの更新時に複数サーバで同期してデータを更新することで問題は解決する。新聞の配送についても、求める情報へのアクセスがいたるところから可能となれば画一的な情報の配信に対する要求は低くなると考えられる。早朝電子ブックへ新聞記事の配信を受けて通勤電車で読むと言った要求が考えられるが、遠隔会議が可能な時代に通勤する価値があるのかどうか疑問である。配送の実時間性要求の小さい大量データを同報する要求は、本当にあるのか。音楽データやゲームソフトウエアの発売開始、同時販売のように配信には実時間性、同時性の要求が少なからず付随すると考えられる。[1]

要求の整理

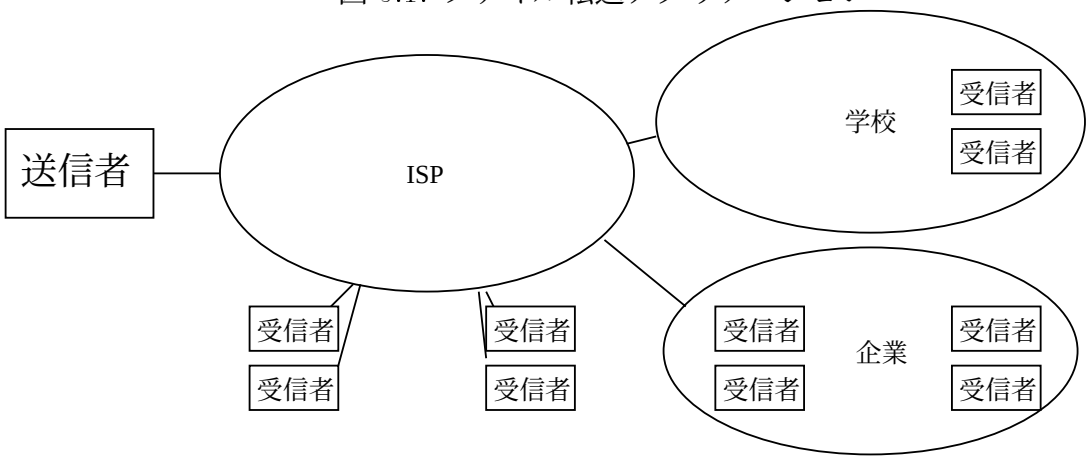
ここで本章で想定するアプリケーションの要求を整理する。

- 2〜3MB のファイルを
- 連続して数個
- 1 時間以内に
- 128kbps 最大帯域を持った回線で常時接続している 100 万台の PC へ
- TCP と公平な帯域共有を必須とする自律ドメインを含む複数の自律ドメインを越えて
- 同報する。
- 受信状況が送信元で確認できる必要がある。

この様子を図示したものを図5.1に示す。IPS に常時接続契約を行なっている受信者、企業内ネットワークに接続されている受信者、学校のキャンパスネットワークに接続されている受信者へ送信者からデータ配送する。

[1]家庭の電子レンジへのレシピ情報の配信も同様である。料理研究家が考案した目新しい献立を配信しないと消費者は見向きもしないであろう。

図 5.1: ファイル転送アプリケーション



5.2 設計

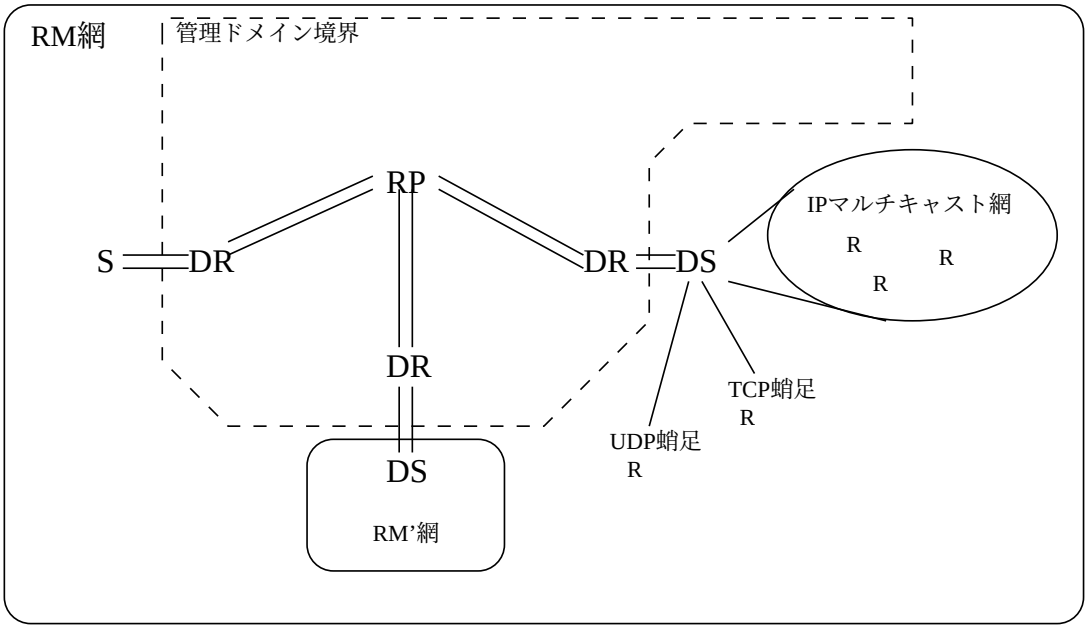
TCP との帯域共有、受信状況確認の必要性、100 万台への同報という要求から設計空間は厳しく規定される。複数の ISP を越えて IP マルチキャストパケットを転送するためには、複数ISP で共通のマルチキャスト経路制御プロトコルを共同で動作させるか、BMGP のようなマルチキャスト経路制御の EGP を動作させる必要がある。さらに、TCP と公平な帯域共有を行なうためには、トランスポートプロトコルが TCP と親和性がある輻輳制御を行なう必要がある。

今回は、図 5.2に示すように既存の高信頼マルチキャストプロトコルを使わない設計を行なった。RM 網は、内部に RM 網を持った再帰的な構造をしている。管理ドメインを越えるための網の構成要素は DR,RP, である。これらは互いに TCP 接続を行ない、双方向のデータを通信する。図中の DS はファイルを中継して下流の受信者もしくは DS へファイルを転送する。DS は IP マルチキャストと損失回復を行なう機構、複数受信者と TCP 接続を行なう機構、複数受信者へ UDP ユニキャストパケットを送付する機構を有する。

5.2.1 中継者の設置

100 万台への送付、受信状況確認の必要性から受信者を論理的に階層的な構造に配置し、委任送信者 (DS) すなわち中継者に受信者もしくは下流の委任送信者への配送を任せる。1 つの中継者が管理する受信者の数の上限を 1000 とすれば要求を満たすためには 2 層以上の構造で最低 1000 の中継者を設置する必要がある。中継者は、ISP だけでなく、顧客企業、団体にも設置する必要がある。その様子を図 5.3に示す。

図 5.2: ファイル転送アプリケーションの設計



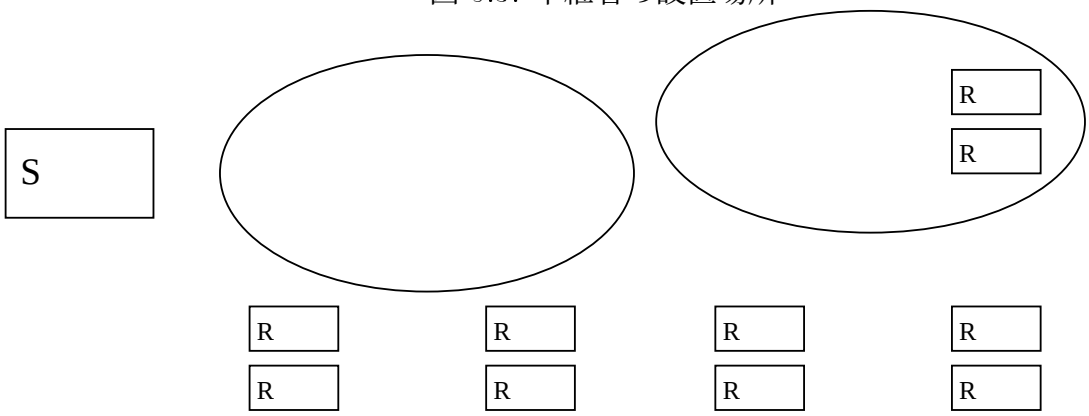
5.2.2 配送網、制御網

中継者への受信者の配送責任の委任、受信状況確認を行なう必要がある。数個のファイル転送動作を 1 セッションとすると受信者が途中参加を行うことができるタイミングは、課金の単位であるファイル転送毎である。このような動作を可能とするため、送信者、中継者、受信者は制御メッセージを交換する。ファイル転送を行なう動作を開始する前には配送網を確立する。一連のセッション動作を開始する前に制御網を確立する。一連のファイル転送動作の流れを図5.4に示す。

5.2.3 非同期転送

ファイル転送は定レートで行なう (一部 TCP 接続を利用しているが)。レートを調整すれば、許容される帯域内で受信者状況確認と連続するファイル転送動作を非同期で実行可能である。この様子を図5.5にしめす。非同期に実施することで 1 セッションあたりの総ファイル送信時間を短くする事ができる。非同期転送を可能にするため、送信者、中継者、受信者は図5.6に示すような構成を採用した。中継者は制御メッセージを受信して、受信者委任情報の受付、ファイル転送の受信開始、ファイル転送の送信開始、受信状況確認メッセージの送付 (一定期間ポーリング)、受信状況報告メッセージの集約といった動作を行なう。受信者から送付される損失回復要求メッセージ、受信者状況報告メッセージは中継者が管理する受信者の数に応じて適宜バックオフされる。

図 5.3: 中継者の設置場所



タの最大長は損失回復の単位となる。データサイズを大きくすると送信者、中継者、受信者アプリケーションのメッセージ処理に関するオーバーヘッドを削減できるが、IP データグラムは最大転送単位(MTU)で断片化されるので、網の損失率が高い場合、結果として不効率な損失回復が行なわれる。

配送網からデータメッセージを受けとった中継者、および受信者は、ヘッダで示されるファイルの特定位置(オフセット)にペイロードデータを転送格納する。ファイルが完全に構成できたら、そのファイルが受信完了状態になる。セッション開設時には、ファイルの転送速度が制御メッセージで通知される。受信者はタイムアウトを契機に未受信データを検出し損失回復する。

5.3 考察

アプリケーション設計を受けて、これより高信頼マルチキャストプロトコルに関連する事項を考察する。

5.3.1 中継者設置基準

本章で見たアプリケーションはTCP 等を用いて設計したがRMTPII を用いて設計することも可能である。双方に共通する枠組は、中継者を設置し受信者を階層的に管理する点である。本節では中継者の配置に関して考察する。

ネットワークトラフィック削減の観点からは、中継者の設置場所は受信者へのマルチキャスト配送木になりうる最小経路木(Shortest Path Tree)上に配置することが理想的である。すなわちルータである。事実、配送木中の高信頼配送はルータがフローを認識し優先的に処理することで実現できる。このアイデアは、IETF の統合サービス作業部会が提案する保証サービスでめざすものと同じである。

さて、本章で設計した中継者は多量のメモリもしくは外部記憶の利用を必要とするアプリケーションである。多量のフローの高速処理処理が要求されるルータへの実装は現実的とは言えない。中継者の配置方法は、静的に設定する方法とセッション毎に動的に配置する方法が考えられる。静的な設定は受信者の増減に応じて手作業で行なう必要があり、複数管理ドメインに設置するような場合は管理者どうしが連絡を取り合いながら行なわなくてはならない。通常、IPS は自らがサービスする内容についての管理コストについては支払うが他のISP と協調してサービスを行なう事に関しては尻込みしがちである。まして顧客企業、団体の管理者は、管理コストをかけたくない。

動的に中継者を設置する方法を考える。まず、アプリケーションの配布方式を考える。配布方法として受信者アプリケーションと中継者アプリケーションを別々に配布する方法と受信者、中継者双方の機能を持った一つのアプリケーションを配布する方法が考えられる。

昨今の計算機能力やLANの帯域の向上から考えて、自ドメイン内のトラフィック最適化のために、受信者アプリケーションが中継者アプリケーションとして動作することが許容されるかもしれない。

特定ドメイン内部の動的な中継者の決定アルゴリズムとして次のようなものがあげられる。

- FQDNでクラスタリング
- IPアドレスでクラスタリング
- RTTでクラスタリング
- ホップ数(TTLを利用して計測)でクラスタリング

送信元からの調査パケットがマルチキャストされ得られた情報を元に上記基準で最適な中継者を選定し送信者が委任する。これらはいずれも次の欠点を持つ。

- 動的に選定された中継者と委任される受信者が同じ自律ドメインであることは保証されない。
- 輻輳リンクや帯域幅の小さいリンクのトラフィック削減を保証するわけではない。

これらの問題を解決するために、いくつかの中継者を静的に設置する方法や受信者から配送範囲を徐々に大きくながら調査パケットを送付して、中継者の候補から適切なものを選定する方法が考えられる[2]。

5.3.2 要求される信頼性の差異

本章で設計したアプリケーションでは、制御網、配送網という2つの配送グループを想定した。制御網はアプリケーションの動作を規定するパケットが配送される。配送網では、アプリケーションが本来配送すべきデータが流れる。

[2]現在、IETFの高信頼マルチキャスト作業部会では、肯定応答ベースの構築ブロックに関する草案について議論中である。その中で配送木の構造を外部ソフトウエアから与えられることを仮定すべきかどうかという議論も展開されている。

それぞれの網の要求について見てみる。制御網のパケットが欠落し回復できなかった場合、アプリケーション動作は停止する。制御パケットの到着の遅れはアプリケーション全体のスループットを低下させる。また、制御網を流れるパケット数は配送網と比較して少ない。

制御網は、配送網と比較して、実時間性、信頼性とも高くする必要がある。許容される帯域はパケット流通量と比較して十分大きいので、パケット単位での冗長化などで信頼性を確保する方法が考えられる。

制御網では配送するコンテンツの暗号鍵の配送を行なうことができる。階層的に受信者を管理するモデルでは暗号鍵の配布を容易に実現することができる。

要求分析、設計を進めると高信頼マルチキャストアプリケーションは複数の性質の異なるマルチキャストセッションにより構成できることが判明する。

5.3.3 想定する受信者の性能

肯定応答ベースのプロトコルを実用化するためには、極端に遅い受信者の扱いなどを規定しておく必要がある。許容する遅延時間を大きくとれば、信頼性が向上することが期待できるが、全体のスループットを低下させる原因になりうる。矛盾するように聞こえるが高信頼マルチキャストアプリケーションの実用には極端に遅い受信者を切捨てる機構が必ず必要になる。もしその機構がないなら、否定サービス攻撃(DoS:denial of service atack)が可能となる。

5.3.4 メッセージの意味

配送網に送付されるデータメッセージは、特定ファイルの特定部分のデータを示す。受信者アプリケーションは、データメッセージを必ずしも送付順序通り受けとらなくても目的のファイルを構成することができる。また、同じデータメッセージを受けとっても問題はない。

このファイル転送アプリケーションはトランスポート層に重複回避や順序保証を要求しない。

図 5.4: ファイル転送の流れ

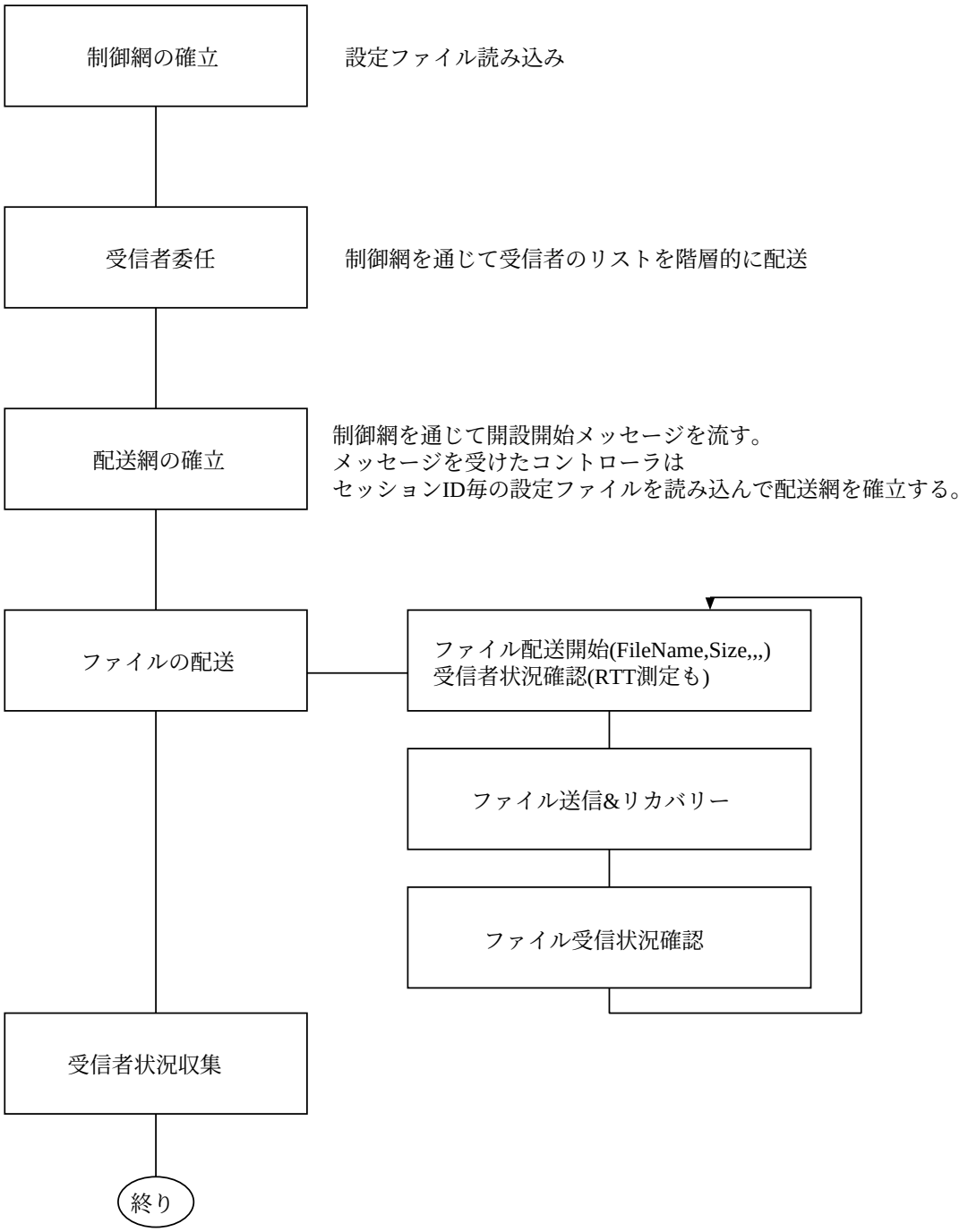


図 5.5: ファイル送信と受信状況確認の同期処理/非同期処理

同期動作

	時刻1	時刻1+a	時刻2	時刻2+a	時刻3	時刻3+a
1階層	ファイル1転送	状況確認1	ファイル2転送	状況確認2	ファイル3転送	状況確認3
2階層			ファイル1転送	状況確認1	ファイル2転送	状況確認2
3階層					ファイル1転送	状況確認1

非同期動作

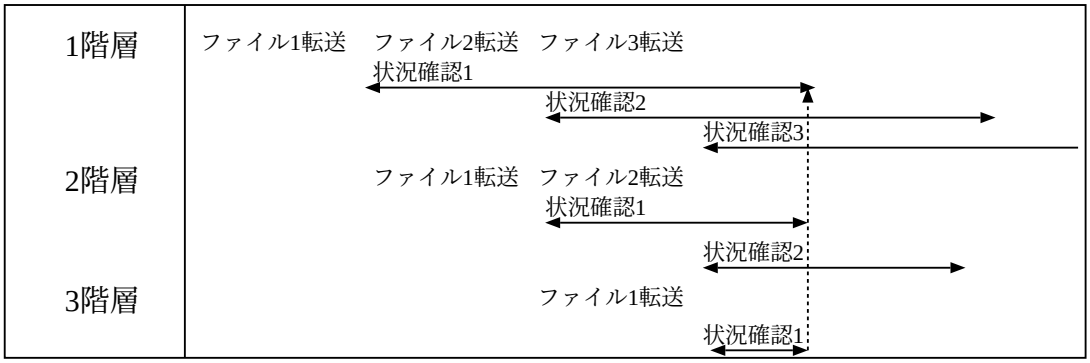
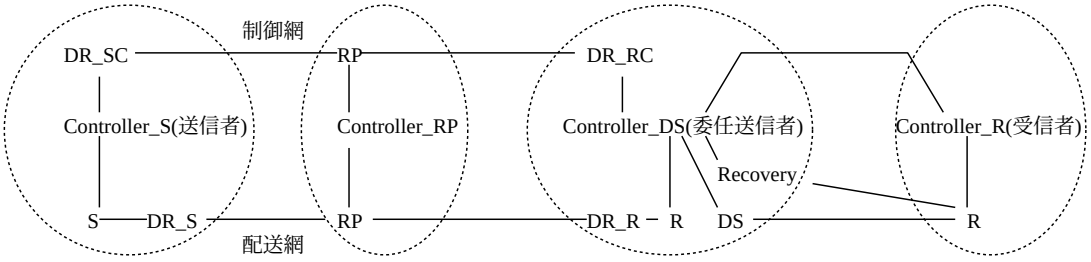


図 5.6: 送信者、中継者、受信者の構造



第 6 章

実時間アプリケーションの実用化の検討

本章では、実時間性の要求の高いマルチキャストアプリケーションの実証実験について紹介する。アプリケーション要求を分析し、設計実装することで、その設計指針や、既存の高信頼マルチキャストプロトコルの改善要求の抽出することを目的に行なった。

配送する対象データとしては、MP3 形式[1]の音楽データ、DV[2]形式のビデオデータを取り上げた。

6.1 実時間アプリケーションの特徴

実時間メディアの転送は、低遅延、低ジッタな配送が求められる。ここで実時間メディアのマルチキャストアプリケーションに関する特徴を見ていく。

6.1.1 同期、非同期、寛容な (isocronous)

ビデオ、音声などのストリームデータ転送は、送信者受信者を完全に同期して行えれば、送受の機構が簡略化でき理想である。受信側でデータを利用するタイミングが送信できる時間と特に関係がない場合、先行的に非同期でデータを送付しておいて後から受信側で再生することができる。転送速度にバラツキがあるネットワークでも最大転送能力、平均転送能力が送信者の送信速度よりも高い場合、同期転送以外の方法でストリームデータを連続転送し、受信側で再生することが可能である。そのとき受信側では有限長のプレイアウトバッファ機構が必要となる。

[1]MPEG1 Audio layer 3

[2]Digital Video format。525-60 形式。フレーム間圧縮はない。

寛容な転送(isochronous transportation)とは、時間に依存するが厳密な同期転送ほど厳しい制約下で行なわれるものではなく、一定の時間範囲の中での行なわれる転送をさす。

我々は、次のような転送機構で寛容なサービスを受ける事ができる。

- 損失発生間隔が一定以上であることが保証されているデータリンク転送
- ATMネットワークの転送
- 統合サービスアーキテクチャの保証サービスが行なわれているインターネットの経路上のパケット転送
- 努力最善型の配送を行なうネットワークで平均損失発生間隔が一定以上(、損失発生率が一定以下)であることが保証されており再送による損失回復を行なう機構と有限長のバッファを有するトランスポート層による転送
- 山村に毎週一度トラックを運転して海産物を運搬する魚屋さんによる配送[3]

6.1.2 RTP、RTCP概説

RTP(Realtime Transport Protocol)は、マルチメディアデータの配送を行なうため開発された。RTCP(Realtime Transport Control Protocol)は、RTPの仕様の一部とされておりRTPのデータ転送を制御、監視するためのプロトコルである。[4]

RTPでは、

- シーケンス番号
- タイムスタンプ機能

を提供し、受信側でリアルタイムストリームの再生することを可能にしている。RTPは、ペイロードを同一視するための拡張アーキテクチャにもとづいており、複数ストリームを同期させて再生させる枠組を提案している。

RTCPは、セッション関連の仕事を支援すること責務として次の4つの機能を提供する。

- サービス品質の監視と輻輳制御
- メディア間の同期

[3]海産物の収穫は安定しており、魚屋さんは風邪をひくことはできない。山村には氷室か冷蔵庫が必要。(腐った魚はいただけない。)

[4]RFC1889でRTPバージョン2が提案標準として公開されている。

- 送信元の識別
- セッション規模の推定

具体的には、損失率やジッタに関する情報を含んだ送信者レポート、受信者レポートを周期的に交換する事でこれらの機能を実現する。[5]

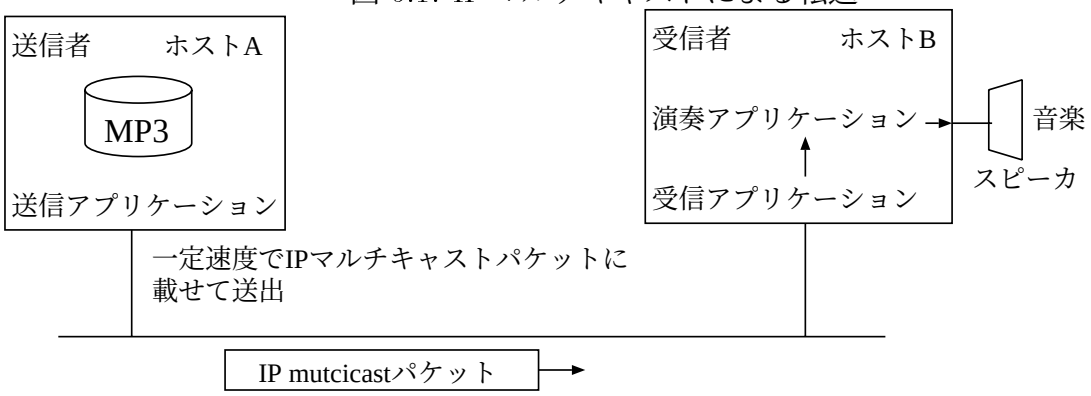
6.1.3 再送による遅延

再送により信頼性を確保するトランスポートを用いて実時間性の要求が高いデータの配送を行なうためには、アプリケーションでのバッファリング機能が必要である。

ここでは、順序保証を行なうトランスポートで再送による遅延発生するメカニズムについて説明する。音声データ、動画データのような実時間性の必要なデータ用いて音声、画像のデータを再生するアプリケーションは、再生するある大きさのデータグラムを一定間隔で音声あるいは画像を再生する周辺機器が接続されているインターフェイスに出力する必要がある。今、図6.1に示すような装置を考える。図中左に示すホストAはローカルディスクに保持している音声データをデータリンクの許容する最大転送単位のデータグラムに分割して一定間隔でで図中右に示すホストBに向けてデータの配送を行なっているとする。(損失率の0のデータリンク上に接続された2台のホストの間でユニキャストでデータグラム指向の転送が行なわれている状況を想定する。) この時ホストBのアプリケーションでは、ネットワークから一定間隔でデータグラムが到着する事が期待できるので、socketから受けとったデータを特にアプリケーション内で蓄積せずに直接音声出力デバイスのディスクリプタに出力する事で正常な音声の再生が期待できる。(ジッタのない理想的な状態) 次に、図6.2を考える。ホストC,Dには再送による損失回復を行なうトランスポートプロトコルが実装されており、上の例と同様にホストCからDへ向けて一定間隔で音声データの配送が行なわれており、ホストC、D間のネットワークではパケットが損失することがあるとする。パケット損失はホストDでパケットに付与されているシーケンス番号の隔たりにより検出され否定応答(NAK)をホストCに返す事で再送要求しホストCからパケットが再送され、損失を回復する。損失検知以後ホストDが受けとったデータパケットは順序保証をするため受信ウインドウで保持され再送パケットが到着するまでアプリケーションに渡されない。すなわち、アプリケーションから観測すれば到着するパケットの間隔は一定ではない。この様子を表現したタイムラインダイヤグラムを図6.3 に示す。

[5] この外に送信元記述パケット、BYEパケット、Appパケットが定義されている。

図 6.1: IP マルチキャストによる転送



演奏アプリケーションが望む時刻に演奏すべきデータが読み出されない場合は演奏が停止する。また、多量のデータを一時に到着すれば、アプリケーションの実現方法によっては演奏アプリケーションの入力バッファでデータが溢れることが予測される。

バッファリング機構

この問題を解決するためにホスト D のアプリケーションでは、バッファリングを行なう必要がある。このバッファの大きさ B(Byte) は、パケットサイズ S(byte) とパケット到着速度 RX_RTE(Packet/Second)、ジッタの大きさ J(Second) とすると

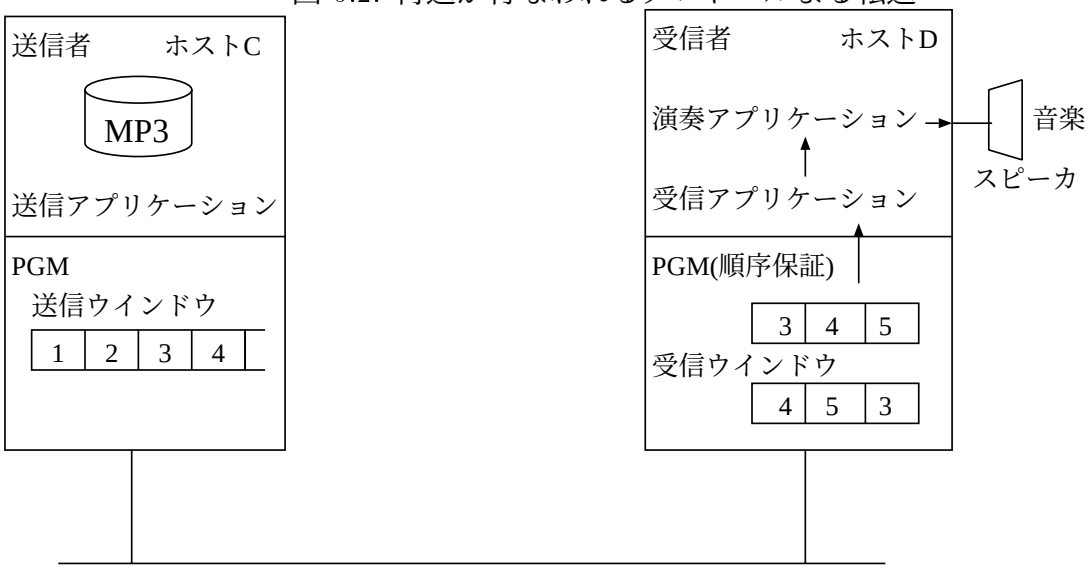
$$B = 2 * (RX\_RTE * J)$$

で表されることが知られている。具体的には、図 6.4で表す通り、演奏アプリケーションと受信アプリケーションの間で、上記の式で表されるデータを一時蓄積し、演奏アプリケーションに対して、一定速度でデータを送出するモジュールを追加する。このモジュールはバッファへ一定量のデータを読み込んむまで、演奏アプリケーションへデータの受渡しを行なわず、メモリへのデータ蓄積を行なう。一定量にデータが達すれば、演奏アプリケーションへのデータの受渡しを実行する。

6.1.4 階層的なバッファリング

実時間アプリケーションの実現のためには様々なバッファリングが行なわれる。ここでは、ファイルに格納されている圧縮された音楽データの再生を行なうアプリケーションを例に発生する様々なジッタと解決方法の関係について述べる。音声データの演奏を行なうアプリケーションでは、ファイルに格納されているのデータフレームを読み込み、圧縮されている音声データの解凍 (decode) を行なう。次にこの音声データをスピーカが接続さ

図 6.2: 再送が行なわれるプロトコルよる転送



れている装置に受け渡す。この音声データの受渡しは、音声再生が途切れず滞りなく再生できるように行なう必要がある。ユーザ空間のアプリケーションが特定装置のデータ書き込みを時刻制御できる精度は、OSのスケジューリングの粒度に制限される。(50〜60Hzが一般的) 同じ現象はシステムコールのレイテンシの揺れとして観測される。このような問題を解決するためには、音声装置のドライバもしくはハードウエアに一定量のバッファリングが必要となる。圧縮されたデータの一つのデータフレームに格納されている音声データの再生時間は、音声データの圧縮率によって異なる。また、データの解凍(decode)に必要な演算時間もフレームごとに異なる可能性がある。このような差異を吸収するためには、演奏アプリケーションで一定量のバッファリングを行なう必要がある。[6]次に、データグラムの配送における到着時刻の揺れについて説明する。インターネットにおけるデータグラムの配送は、最善努力型の転送が行なわれる。データリンクの境界を越えてデータグラムの配送が行なわれる場合、ルータによりパケットが転送される。あるデータグラムがルータの出力キューに留まる時間は、トラフィックの状況に依存する。また、前節で述べた通り、トランスポート層において再送による配送の信頼性向上、順序保証が行なわれる場合、受信者アプリケーションにデータが到着する時刻は、送信者アプリケーションがデータを一定時間で送出しても、一定とはならない。

このように、実時間アプリケーションを実現するためには、様々なレベルでのバッファリングが必要になる。この関係を表6.1にしめす。

[6]例えばmpg123 for FreeBSDの場合、共有メモリを確保したのちforkし共有メモリへの書き込みプロセスと読み込みプロセスが独立して動作するように実装されている。

図 6.3: 損失回復時のタイムラインダイヤグラム

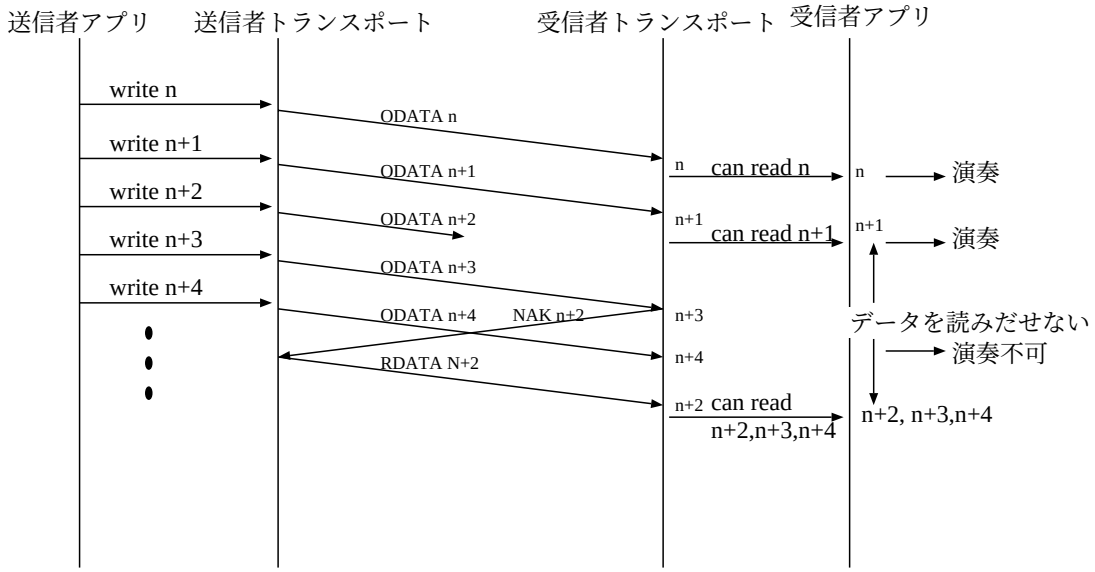


表 6.1: 階層的なバッファリング

階層	問題	バッファサイズ/転送速度
App	圧縮解凍時刻、量の揺れ	圧縮アルゴリズムに依存
4	順序保証された到着時刻の揺れ	$o(RTT)$
3	伝搬遅延の揺れ	o(数百ms)〜$o(s)$
2	OS スケジュール粒度	o(数十ms)

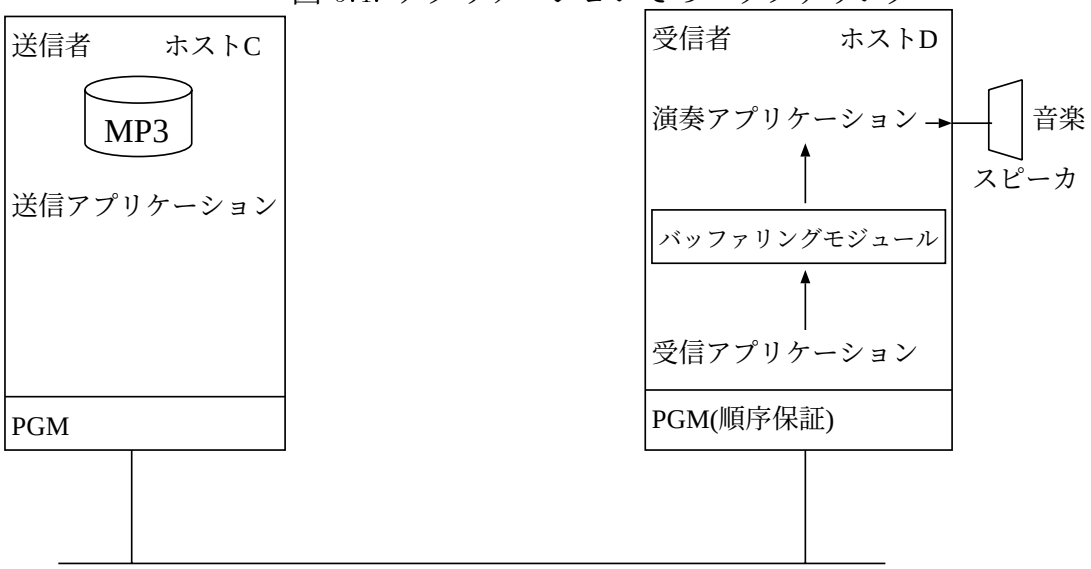
6.1.5 経路設定

IP マルチキャストの配送を行なうためには、ルータが経路制御プロトコルを用いてマルチキャスト配送木を形成する必要がある。経路制御プロトコルは、共有リンクでの転送コストを最小限に抑えるという目的で配送木を形成する。マルチキャストでは広域に分散した受信者が動的にグループに参加、離脱する。[7]複数のマルチキャストグループを使用して輻輳制御を行なう提案もされている。また、フローに対して動的に資源を割り当てる枠組も提案されいる。

配送木の設定は実時間アプリケーションの要求する速度で達成できるべきである。

[7]管理ドメインを越えてマルチキャスト経路情報を交換する方法は標準化の途上である。

図 6.4: アプリケーションでのバッファリング



6.2 MP3 配送実験

無線ネットワークという自然損失が発生する実用環境での高信頼マルチキャストを用いた実時間配送の問題点を顕在化させる目的で音楽データ (MP3 形式) の配送実験を行なった。

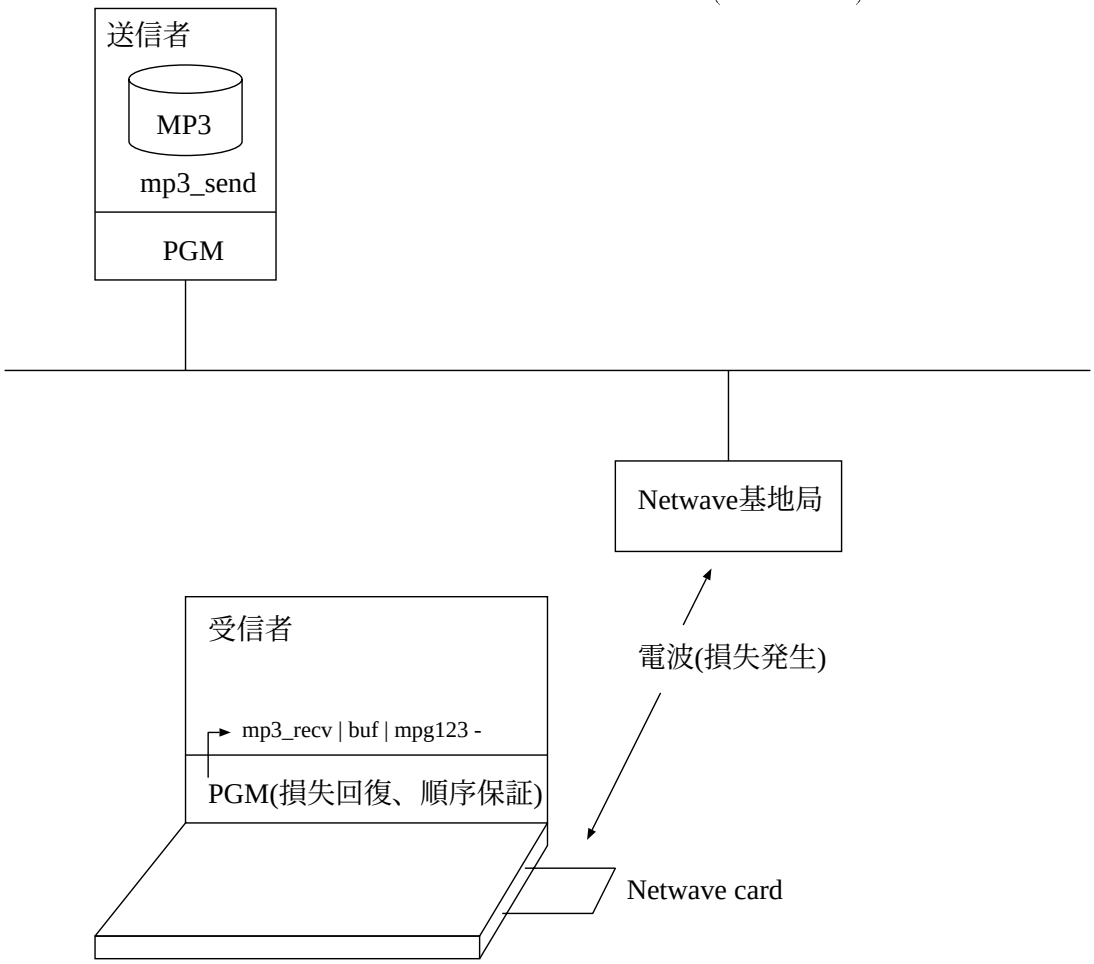
本節では、実験の内容と結果、それに基づく考察内容を記述する。

6.2.1 実験環境

Xircom 社の Netwave を用いた無線ネットワークをデータリンク層として用いて MP3 データのストリーム配信実験を行なった。高信頼マルチキャストトランスポートプロトコルとしては PGM を採用した。実験環境を図 6.5 に示す。

送信者アプリケーション (mp3_send) は MP3 データを 1 フレーム毎 1 パケットで定レートで送信する。無線ネットワークで発生した損失は PGM により回復され順序保証されたデータが受信者アプリケーションでデマルチプレクスされる。受信者アプリケーションは、受信プログラム (mp3_recv)、バッファリングプログラム (buf)、演奏プログラム (mpg123) から構成される。受信プログラムは、PGM から順序保証されたデータがデマルチプレクスされるデータを select システムコールを発行して待っている。デマルチプレクスしたデータは即座に標準出力に出力する。バッファリングプログラムは、環状バッファを実現している。標準入力から読み込んだデータを指定した量だけ、環状バッファに蓄える。その後、select システムコールを用いて環状バッファから標準出力への出力と標準入力から環状バッファへの入力を適宜行なう。演奏プログラムは、標準入力からデータを読み込み、

図 6.5: MP3 配送実験環境 (無線 LAN)



解凍処理を行ったのち、音声出力装置に音声データを書き込む。

送信者アプリケーションと演奏アプリケーションの間で寛容な (isocronous) な配送サービスが提供されなければならない。

ハードウエア環境

本実験で使用した送信者ハードウエアの諸元を表6.2に受信者の諸元を表6.3に示す。

ソフトウエア OS 環境

送信者、受信者とも利用した OS は FreeBSD-3.2R をベースに必要なカーネルパッチを適用したものである。使用したソフトウエアと入手元を表6.4に示す。カーネルパッチ適用後、/sys/netinet/pgm_usrreq.c を修正して、pgm.bandwidth の設定値の上限を 100000000 まで設定できるように修正した。

表 6.2: ハードウエアの諸元1

構成	仕様
CPU	AMD K6-III 400MHz
マザーボード	Freeway FW-T15VGF
チップセット	VIA Apollo VP3
FSB	100Mhz
主記憶	98,304KB
IDE HD	Quantum Fireball CX6.4A
NIC	DEC DE500-AA 21140A(de0)

表 6.3: ハードウエアの諸元2

構成	仕様
CPU	Pentium/P55C 164MHz
チップセット	Intel 82439TX MTXC
FSB	66MHz
主記憶	81,920KB
IDE HD	IBM-DCXA-210000
NIC	Xircom,CreditCard Netwave(cnw0)

6.2.2 受信者、送信者設定

PGMでは、配送の特性を規定するいくつかのパラメータが定義されている。本実験で用いたパラメータとその意味を表6.5に示す。

現在のPGM実装では送信者における送信ウインドウを進める方策は、時間駆動のみがサポートされている。[8]

本実験で用いた設定値を表6.6に示す。

バッファリングプログラムの設定値としては、バッファ容量を320Kバイト、プレバッファ容量を160Kバイトとした。

6.2.3 実験結果

比較のため次の2つの実験を行なった。

[8][17]では送信ウインドウの進め方の方策として、時間駆動の外に、データ駆動、ALFが提案されている。

表 6.4: ソフトウエアの諸元1

構成	仕様	所在地
OS	FreeBSD-3.2RELEASE	http://www.jp.freebsd.org/
PAO	PAO3-19990809.tar.gz	http://www.jp.freebsd.org/PAO/
PGM	kpgm-3.2R-990814.patch	http://www.iet.unipi.it/ luigi/pgm.html
mp3_send,mp3_recv	mp3pgm000129_2131.tgz	lss5-is26:~emuramot/pgm-app/
buf	buf19991227.tar.gz	mailto:thiro@jaist.ac.jp
IP マルチキャスト用アプリ	mp3mc.tar	lss5-is26:~emuramot/pgm-app/

表 6.5: PGM の動作を規定するパラメータ

パラメータ	デフォルト値	意味
pgm.sendspace	150000(バイト)	送信ウインドウの最大サイズ
pgm.recvspace	65536(バイト)	受信ウインドウの最大サイズ
pgm.nak_bo_ivl	4(tick)	NAK をバックオフ時間を算出する基準値
pgm.ncf_to_ivl	5(tick)	NCF を待つタイムアウト値
pgm.rdata_to_ivl	20(tick)	RDATA を待つタイムアウト値
pgm.ncf_retries	6(回)	NCF 待ちで NAK を再発行する最大回数
pgm.rdata_retries	10(回)	RDATA 待ちで NAK を再発行する最大回数
pgm.spm_ivl	15(tick)	SPM の発行間隔
pgm.bandwidth	100000(bit/秒)	送信者の最大転送速度
pgm.odata_lifetime	10(秒)	送信者の送信ウインドウ中に送信済みデータが留まる時間
pgm.pgmcksum	1	チェックサム機能を有効にするか

- IP マルチキャストによる配送

- PGM を用いた配送

について、それぞれ30 秒程度の曲 (500KB) を 3 回送信して、損失率と演奏状態を観測した。結果、IP マルチキャストの配送では、2%の損失を観測し、微小な演奏の途切れを観測した。PGM を用いた配送では、パケットレベルでは12%の損失を観測したが、いずれの損失についても回復動作が行なわれ途切れない演奏を観測した。

結果を表6.7にまとめる。

表 6.6: MP3配送実験時のパラメータ

パラメータ	送信者	受信者
pgm.sendspace	500000	デフォルト値
pgm.recvspace	デフォルト値	デフォルト値
pgm.nak_bo_ivl	デフォルト値	デフォルト値
pgm.ncf_to_ivl	デフォルト値	デフォルト値
pgm.rdata_to_ivl	デフォルト値	デフォルト値
pgm.ncf_retries	デフォルト値	デフォルト値
pgm.rdata_retries	デフォルト値	デフォルト値
pgm.spm_ivl	5	デフォルト値
pgm.bandwidth	1000000	デフォルト値
pgm.odata_lifetime	デフォルト値	デフォルト値
pgm.pgmcksum	デフォルト値	デフォルト値

表 6.7: MP3配送実験の結果

	IPマルチキャストによる配送	PGMを用いた配送
損失率	2%	12%
演奏	途切れる	途切れない

6.2.4 考察

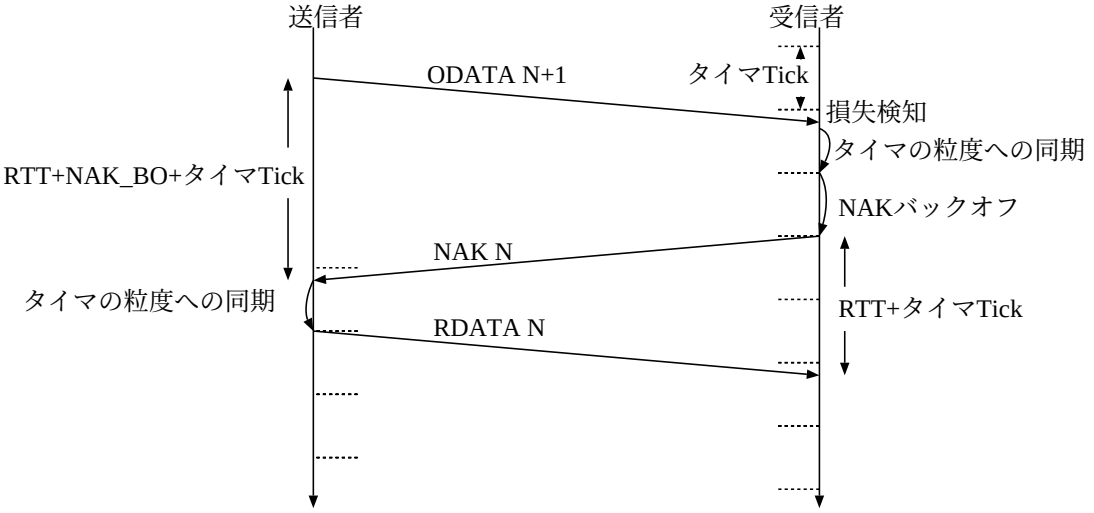
本実験で得られた知見と顕在化した問題点について述べる。

送信者資源所要量、ネットワーク距離、損失率

送信者では、受信者からの否定応答に答えるために送信済みデータをメモリ上に保持している。送信者に必要なメモリ資源の総量は、受信者とのネットワーク距離、パケットの損失状況と深い関連がある。

ここでは、本実験における送信者資源所要量、ネットワーク距離、損失率の関係について述べる。本実験で用いたPGM実装では、タイマ起動のアーキテクチャが採用されている。否定応答によって駆動される再送パケットの送出は、待ち行列に蓄えられ、定期的に起動されるタイマー処理によって送出動作が行なわれる。この様子を図示したものを図6.6に示す。具体的には、この図は受信者が損失を検知しNAKを送信者へ送付する様子を示している。シーケンシャル番号が$N+1$番のODATAを受けとった受信者は、損失を検

図 6.6: NAK,NCF,RDATA 転送のタイムラインダイヤグラム



知し、NAK パケットの送出をスケジュールする。タイマ処理ルーチンはバックオフ時間をカウントダウンし、これがゼロになれば送出動作を行なう。

NAK や NCF の損失がない場合に、送信者が確保すべき送信ウインドウの量について考える。図 6.6から明らかなように、送信ウインドウに必要な S(バイト) は、送信速度を TX_RTE(バイト/秒)、NAK バックオフ時間を N(秒)、カーネルタイマ時間を T(秒)、送信者と受信者の RTT を RTT(秒) として、

$$S = (N + T + RTT) * TX\_RTE$$

で表される。

NAK,NCF,RDATA が損失することを想定して、受信者では NAK 再送をバックオフしている。それぞれの損失があった場合でも、送信者が RDATA の再送ができるように、するための上記の待ち時間 (式右辺の積演算の左項) に対応する NAK のバックオフ間隔を加える必要がある。

NAK バックオフ間隔の計算は、パラメータを nak_bo_ivl、$[0, 2^{31} - 1]$ の乱数を発生させる関数を random() として次のように定義されている。

```
#define SET_BACKOFF (nak_bo_ivl / 2) + (random() % nak_bo_ivl)
```

$RTT < nak\_bo\_ivl$ であると仮定すれば、n 回のバックオフを想定したときに送信者が確保すべき送信ウインドウの量 S_n は、NAK バックオフの平均時間を N_{adv} として次式で表される。

$$S_n = n * (N_{adv} + T + RTT) * TX\_RTE$$

パケット損失の生起を独立事象としてモデル化すれば、n 回連続でNAK もしくはNCF もしくはRDATA が損失する確率は、パケット損失率を $loss$ として $loss^n$ で表される[9]。

途中参加

一連の音楽が演奏されているセッションに途中から参加してその音楽データを受信、演奏させたいという要求が考えられる。そのような状況で途中参加する受信者が取ることができる方策は次の2つが考えられる。

- 参加時点で、受信できるデータから再生する。
- 参加時点で、再送要求できるデータの再送要求を行なう。

後者の方策を採用するためには、再送要求をしたNAK パケットが送信者に届いた時に目的データが送信ウインドウに存在すべきである。また、システム設計時には、見込まれる途中参加数とパケット流通量とネットワークの配送能力を考慮する必要がある。

PGM ではLate join オプションが定義されている。送信者は、ODATA,SPM,RDATA 送出時に、再送要求可能な最小のシーケンシャル番号を記載したオプションヘッダを付与してパケットを送出する。

途中参加を必要とするアプリケーションの例としては、課金の単位を曲単位としている音楽配送などが考えられる。送信者は送付中の曲のデータすべてを送信ウインドウに蓄えている。これは途中参加する見込み数に対して利用可能な帯域が十分大きい場合有効な手法と考えられる。[10]

求められる通信のサービス

すべての経路で統合サービスアーキテクチャの保証サービスを受けた場合、送信者アプリケーションが契約以下のレートで送付すれば、搬送経路上での損失は発生しない。受信者アプリケーションに固定長のプレイアウトバッファがあり安定して動作していれば、損失回復の機構は必要ない。再送要求が行なわれるのは途中参加を契機とするものに限られる。

制御負荷サービス配下で動作した実績のある適応型アプリケーションは、次のセッションでも正常に動作することが期待できる。送信者が送出するレートで制御負荷サービス

[9]特定フローを一般トラフィックが流れるインターネットを用いて転送を行なった時の損失について妥当なモデル化の検討は研究の対象となりうると考える。

[10]ただし、受信者が受信者が音楽データを再生できるタイミングは、受信者の間でばらつく。(同時性に関する検討は別途必要となる。)

を受ければ、送信者が送出するパケットは落ちにくい。落ちたパケットは受信者で検知されて、否定応答が送付される。送信者が否定応答を受けて送信するRDATAは、定レート送付されているODATAに加えて送付される。すなわち超過した分のパケットは制御負荷サービスの恩恵を受けず最善努力型の転送が行なわれる。

制御負荷サービスで契約の範囲内の送信レートでパケットを送付しているときのパケットの落ちにくさと最善努力型転送の品質は、プロバイダの経営戦略で決定される[11]。

制御負荷サービス配下でPGMによる転送を行ない演奏アプリケーションが寛容な(isocronous)サービスを受けるには、再送により発生する遅延を吸収するために十分なバッファをアプリケーションで確保する必要がある。

制御負荷サービスは、ある日突然、他からの流入トラフィックによるバースト負荷により、ODATAの損失発生が大きくなったり、遅延が極端に伸びたりしないことを保証するサービスである。制御負荷サービス配下で、高信頼マルチキャストプロトコルの設定とアプリケーションのバッファの設計と行なう事で、アプリケーションの動作品質を定量的に定義することが可能となる。これは努力最善型の配送を行なうネットワーク上では不可能であったことである。

暗号化と鍵配送

暗号キーを交換を拡張して多数の受信者に鍵を渡しマルチキャストを行なうホストグループの動的なメンバーシップ管理を行なう適切な方法は、まだ見つかっていない。

課金の単位を曲単位として、音楽ストリーム配信を行なうサービスを行なうためには、1曲の演奏の間に次の曲の暗号鍵を配送する必要がある。鍵の配送をPGMのような受信者起動の高信頼マルチキャストプロトコルで効率的に行なう方法は現在考案されていない。

BGMのような連続演奏

本実験で示したアプリケーションは、1曲の音楽データの配送を行なう。受信者は配送されるストリームを一定量プレバッファしてから演奏を開始する。

現在、有線放送が行なっているような連続して音楽を配信するサービスを考える。送信者が送出する速度と受信者の演奏アプリケーションが演奏する速度に際がある場合問題が生じる。送出速度が演奏速度にくらべて早い場合、バッファリングモジュールが確保しているメモリはやがて音楽データフレームでうめられ、やがてデータが損失する。また、送出速度が演奏速度に比べて遅い場合、バッファ中のデータはすべて取り出され無音期間

[11]ユーザとして、なんとも提供品質を監視しにくいサービスである。ユーザとしては、帯域幅をレポートするtracerouteのようなツールが望まれる

が生じる。このような、問題を解決するためには、演奏の切れ目を設定し、送信者、受信者のアプリケーションの間で同期をとる必要がある。具体的には、曲の切れ目などに、送信者がパケットを配送しない時間帯を作り、受信側アプリケーションが無音期間を調整して、バッファ中のデータを一定の範囲内に収める工夫をする必要がある。

6.3 DVTS配送実験

マルチメディア情報の高信頼マルチキャストによる配送における問題点を明確化させる目的で、ビデオカメラで撮影した映像、音声データをリアルタイムでマルチキャストする実験を試みた。ホームビデオカメラで撮影したDV形式の映像、音声は送信者によりPGMでマルチキャストされる。受信者は損失回復を行なったストリームをモニタで再現する。

6.3.1 DVTSについて

DVTSはDV形式のIEEE1394フレームをIPデータグラムにのせてインターネット上で配送するシステムである。[12] DV形式は6.3mm幅の小型カセットテープを用いた大衆ビデオカメラのために設計された形式で、音声データ、画像データ、制御データを画面フレーム単位で管理する[22]。そのため画面フレームのコマ落しなどを容易に実現できる。DVTSは、画面単位のコマ落し機能や、画面フレーム、音声フレーム単位のバッファリング機構を有しネットワーク伝送帯域にあった配送を行なう事ができるシステムである。

フレームレート制御機構

一画面あたり525走査を行ない一秒間あたり29.97フレームの描画を行なうNTSC規格のDVストリームを直接IPデータグラムにのせて配送すれば、30Mbps以上の帯域を必要とする。DVTSの送信者は、画面フレーム、音声フレーム単位で送信を抑止する機構をそなえており、設定するフレームレート(何画面フレームごと伝送を行なうか)で配送を行なうことができる。またDVTS送信者は、音声フレームの転送は抑止せず画面フレームの転送のみ抑止する機構を持っている[13]。送信者の状態遷移図を図??に示す。

設定するフレームレートと利用するネットワーク帯域の関係を表6.8に示す。数値はUDPデータグラムとして送付するときの値である。

[12] http://www.sfc.wide.ad.jp/DVTS/

[13] http://www.sfc.wide.ad.jp/ akimichi/pv2000/

図 6.7: DVTS送信者の状態遷移図

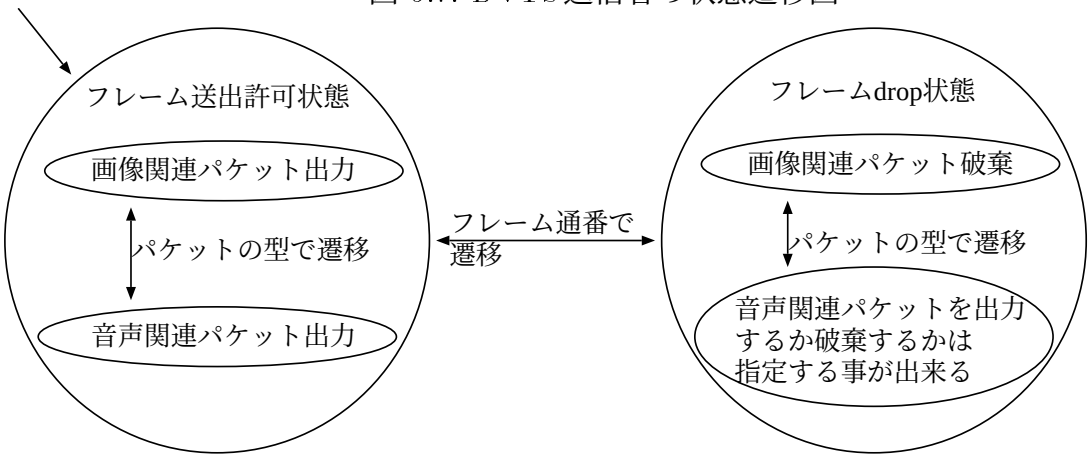


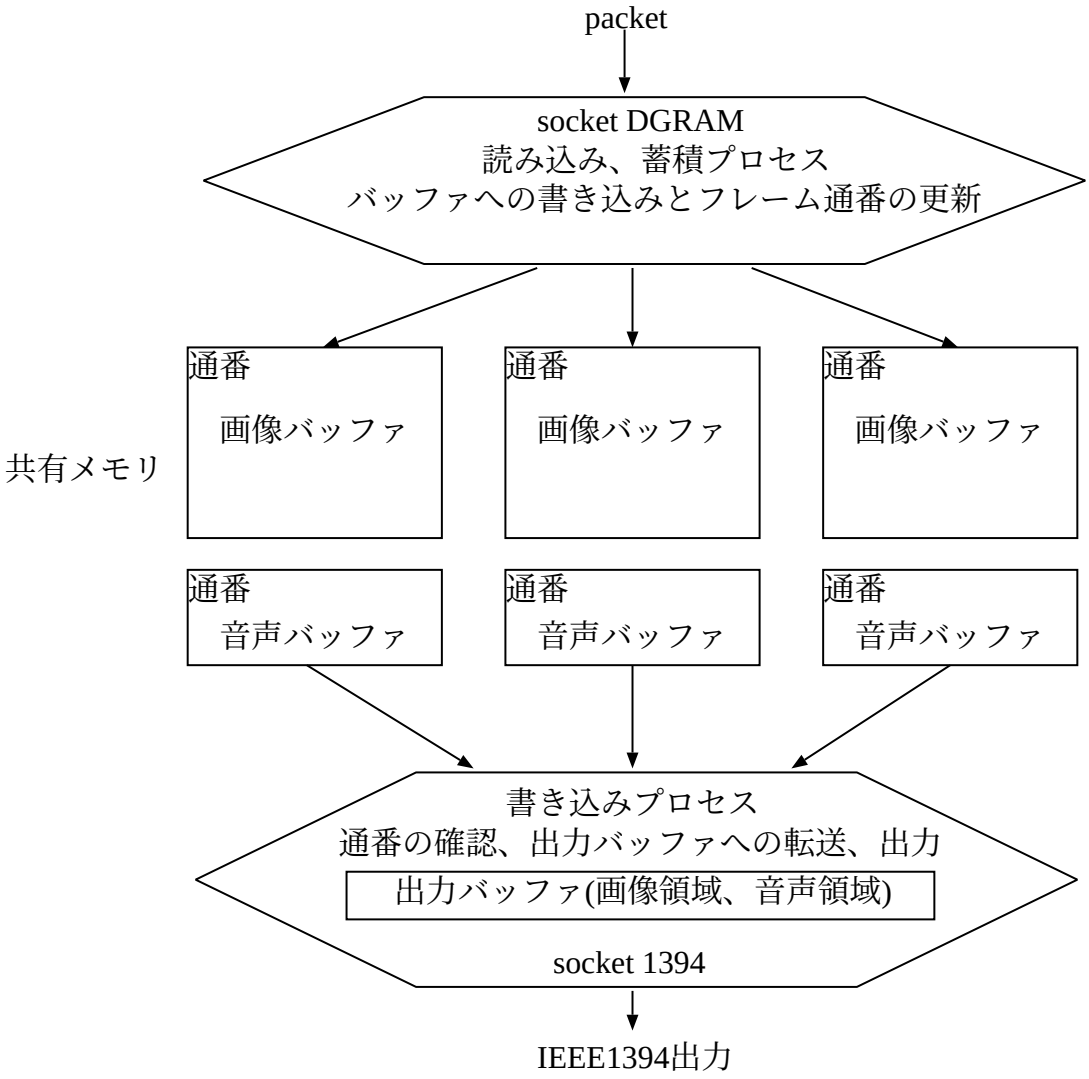
表 6.8: DVTSの消費帯域

設定フレームレート	IPv4上での転送時(Mbit/s)	IPv6上での転送時(Mbit/s)
1/1	30.47	31.70
1/2	15.72	16.83
1/3	11.48	11.84
1/4	9.01	9.33
1/5	7.54	7.83
1/10	4.74	4.87
1/20	3.26	3.39
1/30	2.79	2.90

バッファリング機構

DVTSの受信者では伝搬遅延のジッタを吸収するためのフレーム単位でのバッファリング機構を導入している。映像フレームと音声フレームを別々に管理し、再生する時に合成することで、映像フレームの間引き転送が行なわれる場合でもフレーム単位で整合性のある画面の再生が可能となっている。受信者のプロセスは共有メモリに、映像用、音声用のバッファを確保したのち、forkして、socketからデータを読み込み共有メモリ上のバッファに書き込むプロセスと共有メモリ上のフレームデータを読み込み合成して、IEEE1394の装置に書き込むプロセスとに別れて動作する。これにより遅い事が知られているselectシステムコール[1][2]を使わずに入出力ストリームを協調的に動作させることを実現している。この様子を図6.8に示す。

図 6.8: DVTS のバッファリング



6.3.2 実験環境

DVTS による動画像、音声のストリームの高信頼マルチキャストプロトコルを利用した配信実験を行なった。高信頼マルチキャストトランスポートプロトコルとしてはPGM を採用した。実験環境を図6.9に示す。

損失を発生させる機構

本実験では、同一データリンクを用いて配送実験を行なった。実験系として必要な損失を発生させるために受信者に損失を発生させる機構を導入した。

受信者のカーネルコードの/sys/netinet/ip_input.c に乱数を用いて、500〜550 パケット

図 6.9: DVTS配送実験環境(PGM)

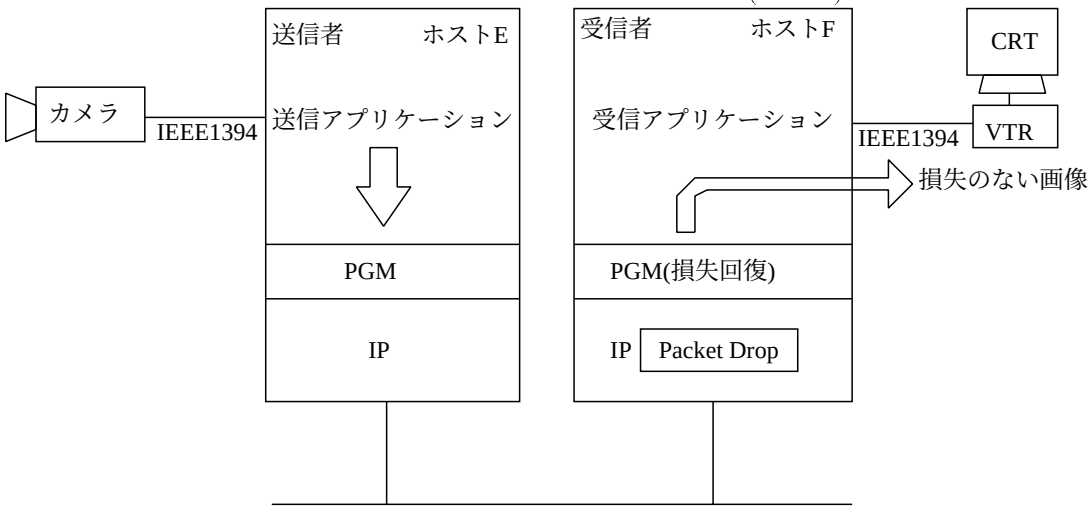


表 6.9: ハードウエアの諸元3

構成	仕様
CPU	Pentium II 266MHz
チップセット	Vendor=8086 device=7180
FSB	66MHz
主記憶	131,072KB
IDE HD	IBM-DHEA-36481
NIC	DEC DE500-AA 21140A(de0)
IEEE1394	Adaptec AIC-5800

受信ごとに1〜3つのパケットを落すコードを挿入した。

ハードウエア環境

本実験で使用した表6.2の諸元のハードウエアを送信者として利用した。受信者の諸元を表6.9に示す。

送信者のIEEE1394ディバイスカードは、Texas Instrument社のPCILynxを利用した。ビデオカメラはSONY社のDVR-TRV5を用いた。

ソフトウエアOS環境

送信者、受信者とも利用したOSはFreeBSD-3.2Rをベースに必要なカーネルパッチを適用したものである。使用したソフトウエアと入手元を表6.10に示す。カーネルパッチ適用

表 6.10: ソフトウエアの諸元2

構成	仕様	所在地
OS	FreeBSD-3.2RELEASE	http://www.jp.freebsd.org/
PGM	kpgm-3.2R-990814.patch	http://www.iet.unipi.it/ luigi/pgm.html
DVTS	DVTS-0.2.0pgm.tar	lss5-is26:~emuramot/pgm-app/
IP マルチキャスト用アプリ	DVTS-0.2.1.tar.gz	http://www.sfc.wide.ad.jp/DVTS/

表 6.11: DVTS配送実験時のパラメータ

パラメータ	送信者	受信者
pgm.sendspace	80000000	デフォルト値
pgm.recvspace	デフォルト値	5000000
pgm.nak_bo_ivl	デフォルト値	1
pgm.ncf_to_ivl	デフォルト値	デフォルト値
pgm.rdata_to_ivl	デフォルト値	デフォルト値
pgm.ncf_retries	デフォルト値	デフォルト値
pgm.rdata_retries	デフォルト値	デフォルト値
pgm.spm_ivl	2	デフォルト値
pgm.bandwidth	90000000	デフォルト値
pgm.odata_lifetime	1	デフォルト値
pgm.pgmcksum	デフォルト値	デフォルト値

後、/sys/netinet/pgm_usrreq.c を修正して、pgm.bandwidth の設定値の上限を 100000000 まで設定できるように修正した。

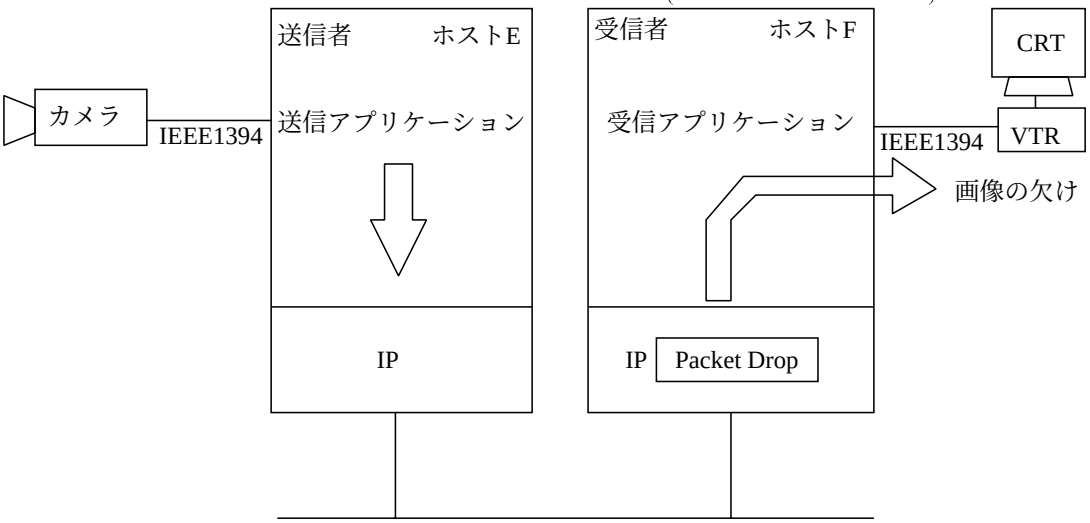
6.3.3 受信者、送信者設定

本実験で用いた設定値を表6.11に示す。

受信者では共有メモリを多量に用いた実験を行なったためカーネルコンパイル時に下記の設定を行ない取得できる共有メモリを量の上限値を増やした。

```
options         "SHMMAX=(SHMMAXPGS*PAGE_SIZE+1)"
options         SHMMAXPGS=18000
options         SHMMIN=2
options         SHMMNI=33
```

図 6.10: DVTS配送実験環境 (IP マルチキャスト)




```
options         SHMSEG=9
```

また送信者では多量の mbuf を消費するため以下の設定をカーネルに対して行なった。

```
sysctl -w kern.ipc.maxsockbuf=95000000
```

6.3.4 実験結果

比較のため次の2つの実験を行なった。

- IP マルチキャストによる配送 (図6.10参照)

- PGM を用いた配送 (図6.9参照)

について、それぞれパケット損失がない状況、パケット損失が550〜650パケット受信中、1〜3パケット発生する状況で行なった。

フレームレート $\frac{1}{30}$ と $\frac{1}{10}$ での実行について、画質の状況、音声の途切れ、ノイズの発生状況を観測した。

結果、IP マルチキャストの配送では、損失がある場合画面上にブロック上の乱れが観測され、音声に雑音が観測された。PGM の配送では、画面は乱れずは観測されず、雑音のない音声が観測された。また、フレームレート $\frac{1}{30}$ の転送を損失のある状況で行ない20分以上の連続再生を行なえる事を観測した。(RDATA が6回連続で損失しなかった)

結果を表6.12にまとめる。

表 6.12: DVTS配送実験の結果

フレームレート	損失	IP マルチキャストによる配送	PGM を用いた配送
$\frac{1}{30}$	なし	画像、音声とも乱れ無し	画像、音声とも乱れ無し
$\frac{1}{10}$	なし	画像、音声とも乱れ無し	画像、音声とも乱れ無し
$\frac{1}{30}$	あり	画像、音声とも乱れ有り	画像、音声とも乱れ無し
$\frac{1}{10}$	あり	画像、音声とも乱れ有り	画像、音声とも乱れ無し

6.3.5 考察

DVTSは画像と音声、2つのメディアのストリームを広帯域を利用して配信し、受信者でこれらのデータを同期させて再生するアプリケーションである。本実験を通じ、高信頼マルチキャストを用いてこれらを配送するに際し顕在化した問題点をここで記述する。

ALFの必要性

DVTSの実現方式について考える。DVTSの目的は受信者が画像、音声が同期して再生できることである。送信者は、ビデオカメラから取得した情報を、マルチキャストグループに送信しつづけ、受信者は受信したデータでDV形式のフレームを構成しビデオ再生装置に書き込み続ければよい。本節では、このような処理を容易に行なう枠組としてALFが有効である事を説明する。

DVTSのような、アプリケーションの設計指針として次の2つが考えられる。

- 受信者でフレーム境界を判別する方法
- 送信者でフレーム境界を判別する方法

前者の場合、送信者はビデオカメラから取得した情報を即座にマルチキャストグループに送出する。後者の場合は送信者はフレーム境界を判定してフレーム境界の識別を受信者で行なえるようにパケットのヘッダに情報を付加してマルチキャストグループに送出する。

一見どちらのアーキテクチャを採用しても、大差ないように思える。しかし、これらの間にはソフトウエアの生成発展と言う観点から差異がある。以後に論拠を列挙する。

- デバッグポイントの確保
  私は、実時間アプリケーションの複雑性の回避のためフレーム境界の判定は送信者で行なうべきであると考える。実時間性プログラムのデバックや、時間的な因果率の崩れに起因するシステムバグの原因追求作業は、現象の再現が困難なため難しい

とされている。具体的には、今回扱ったリアルタイムストリーム同期ずれのデバッグ作業があげられる。パケットの到着時刻と受信ウインドウメモリ状態、アプリケーションのプレイアウトバッファの状態を忠実に再現し解析を進める必要がある。

作業を複雑化している要因は時系列的解析を必要性である。すなわち、プログラム一つ一つの記述に論理的矛盾はないのに、システム全体として異常な動作を行なっている原因は、システム状態を変える事象の生起順序にある。これが設計者、プログラマの意図する範囲を越えため深刻な事態として観測される。

このような問題を緩和するためには、問題を分割して複雑性を緩和することである。本実験に関して言えば、フレーム境界の判定部分を送信者で行なう事でデバッグの容易性が向上する。理由は次の通りである。

- 送信者プログラム完成時点でテストができる。
  受信者でフレーム境界の判定を行なっている場合、送信者受信者両方のコーディングが終了した時点ではじめてテストが可能となる。送信者プログラムで行なう事で、フレーム切替えのタイミングに問題がないかどうかといった問題をテストできる。[14]人間の誤りがプログラムコード量に比例して潜入するとすれば、デバッグポイントが適当に確保されている事は重要である。[15]

- RTP 処理の容易性
  本実験のように受信者で画像、音声といった複数ストリームを合成してコマ落しのような処理を行なうためのプロトコルとしてRTP がある。

  DVTS において、RTP ヘッダを用いてフレーム境界を送信者から受信者に伝達する方法は次の 2 つが考えられる。

  - RTP ヘッダのシーケンシャル番号欄を用いる
    フレームが切り替わるまで、同じシーケンシャル番号を付与する。タイムスタンプ欄は、データフレーム読みだし時刻を記録する。
  - RTP ヘッダのタイムスタンプ欄を用いる
    フレームが切り替わるまで、同じタイムスタンプ値を付与する。シーケンシャ

[14]つまらないことに見えるかもしれないが、現場のプログラマは、ビデオカメラのパケット送出タイミングに問題がないのかどうかという問題と自らが作ったアプリケーションのフレーム境界判定に問題がないかどうかを切りわける必要があることに注意願いたい、プロトコル普及初期の商品やそれを駆動するディバイスドライバ、多量の資源を確保する時の OS 動作に問題がないとは誰も言い切れない。

[15]潜んでいる誤りの数が増えれば、デバッグの時間は急速 (指数関数的に) に増える。(私の感想であるが、、、)

表 6.13: ADU粒度とその特徴

ADU	小さい	大きい
パケットオーバーヘッド	×	○
再送単位の細かさ	○	×
実時間制御の細かさ	○	×

ル番号はパケット送出毎に増やす。

前者の場合、受信者はビデオカメラからデータフレームを読みだしたタイミングを再生できる。後者の場合、送信者が指定するフレーム切替え時刻を再現できる。[16]

いずれの方法を採用するかは、アプリケーション要求に依存する。

いずれにしても、RTP処理を容易に行なう枠組は必要とされている。このようなアプリケーションはフレーム単位での到着の信頼性の向上、順序保証をトランスポート層に要求する。

- DVTSのADU候補
  DVTSにおいて考えられるADUの候補はつぎのようなものが上げられる。
  - 同期する画面フレーム+音声フレーム
  - 画面フレームと音声フレーム別々
  - IEEE1394パケット単位

  これらはアプリケーションの要求により選択することができる。ADUの粒度による利点および欠点は表6.13にしめす。

- 音声優先などフレーム単位の処理の追加の容易性
  画面フレームをコマ落しで送信して、音声フレームをフルレートで送信することでアプリケーションが消費する帯域を節約する方法が考えられる。送信側に、このような処理を追加する次の2つの場合について考える。
  - 受信者でフレーム境界を判別する方法
  - 送信者でフレーム境界を判別する方法

[16] DVTSでは0.2.0から後者の方策が採用されている。

前者の場合、送信者に新たに送信データグラムの音声か画像かといった種別を識別する仕組みと送信を抑止する仕組みを追加する必要がある。後者の場合、送信者に音声フレームの送信を抑止する仕組みのみを追加すればよい。

このように、いくつかあるアプリケーションの要求を整理すれば、後者の方策がより望ましい方策であると判断される。

- 緊急通信によるバッファ中の放送禁止用語の削除処理など受信バッファを前提としたアプリケーション設計
  インターネットは統計多重の効果により投資内容の割に利用者が満足できる品質の通信が確保できると認識されているネットワークである。再送により損失回復を行なったトランスポート層を用いたアプリケーションででは、バッファリングが必要となる。実時間性が要求されるアプリケーションでは低遅延な再生が要求される。しかし、一定時間の遅延を許容すれば、高信頼な画像ストリームのマルチキャスト配信も可能である。

  現在、電波媒体を用いた放送では生放送番組の放送時に送信局側で放送禁止用語の削除を行なうために一定時間蓄積してから放送を行うことがある。[17]あるいは、野球中継などでは、ホームランシーンなどを即座に再放送できるように一定時間蓄積している。

  インターネットを用いた高信頼動画像配送においては、一定時間のバッファリングが必要である。しかし、受信者側のバッファを送信側から制御できるような機構が実現できれば一定の利用用途は想定できる。

  このようなアプリケーションは、トランスポート層に帯域外データ [2] のサポートを要求する。

- MPEG4 のコンテント符合化等、配送信頼度の異なる一連のストリームデータ配送への柔軟な対応
  動画圧縮の形式である MPEG4 では、コンテント符合化に対応している。画像中の特定オブジェクトの動作を信頼度高く配送するといった要求に容易に対応するための枠組は必要とされている。符合化形式の進化とともに、より繊細な信頼度の異なる配送網の使いわけが必要とされることが予測される。

[17] 米国のバスケットボール中継のヒーローインタビュー時など

高速化の技法

DVTSは、フルレート転送を行なうと35Mbps程度の帯域を消費するアプリケーションである。損失回復を行なうために必要なメモリサイズはRTTが同じなら最大転送速度に比例して大きくなる。ここではアプリケーションの高速化で用いた技法について記述する。

- selectシステムコールの回避
  selectシステムコールは遅いことで知られている[2]。DVTSでは共有メモリいてselectシステムコールを回避する方法を採用している。

- カーネル中の設定値の調整
  PGMの送信ウインドウを保持するためにカーネル中のmbufを利用している。100バイトを越えるデータをmbufデータを確保する時には、クラスタと呼ばれる外部バッファ領域が確保される。本実験を実施するためには、次のようにして最大割り当て可能クラスタ数を増加させる必要がある。

  ```
  sysctl -w kern.ipc.nmbclusters=設定値
  ```

DROPオプション

DVTSの送信者は、音声フレームの送信は抑止せず、画面フレームの送信のみを抑止する機能を持っている。PGMでは、配送するパケットの損失が検知されても再送要求しないことを通知するDROPオプションが提案されている。再送要求の必要のないパケットの次に送付するパケットに再送要求が必要な寸前のシーケンシャル番号を記載して送付する。受信者は、シーケンシャル番号の隔たりで損失を検知するが、Dropオプションが付与されている場合は、記載されている再送要求が必要な寸前のシーケンシャル番号より大きなシーケンシャル番号の欠落については否定応答を出さない。

この場合、送信者アプリケーションがデータ送付時に否定応答の必要がない旨をトランスポート層に通知して送信命令を発行する。

送信者でフレーム境界の判定を行ない音声フレーム、画像フレームの判定を行なっていれば、送信時に否定応答の要否を指定することは容易である。

タイムスタンプオプション

PGMでは、アプリケーションが必要とする時間までにパケットが到着する見込みがない場合、受信者が否定応答の送付を抑止するタイムスタンプオプションが定義されてい

る。実時間処理を行なうアプリケーションではRTPヘッダにタイムスタンプを記載してデータを配送する。

アプリケーションレベルからトランスポート層に対してより詳細な指示をすることで否定応答、再送パケットのトラフィックを減らす事ができる。

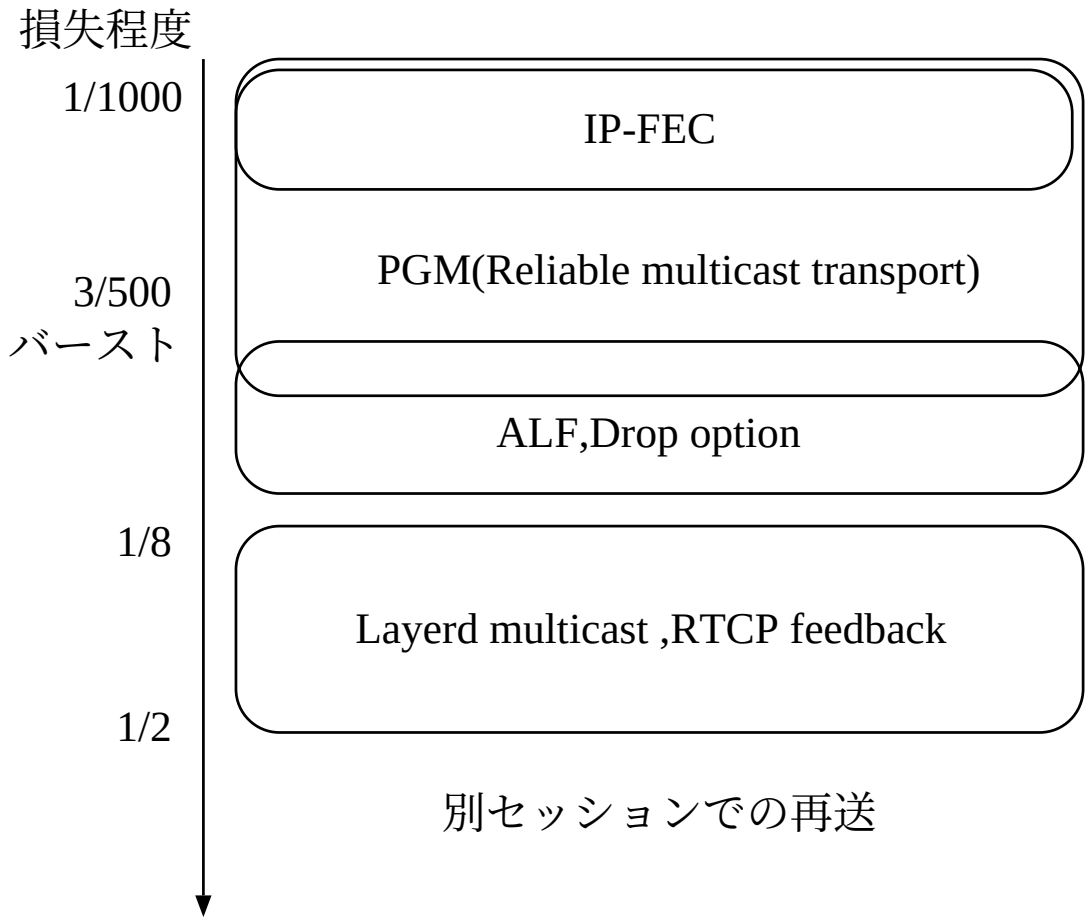
Dropオプション、タイムスタンプオプションとも本実験で用いたプロトコルスタックでは、実装されていない。

損失率と回復の方法

アプリケーション要求を満たすためには、単一プロトコルの強化拡張のみで対応することは最善とはいえない。損失率が小さく、バースト損失がないような場合は、3つのパケットを4つのパケットに冗長符合化するIP-FEC[23]などで十分である。しかし、連続で2つ以上のパケットが損失するような状況では、PGMのような再送ベースの損失回復機構が必要となる。さらに損失が多くなる場合は、(輻輳などが原因と考えられるが)階層符合化とRLCの枠組や流量制御、輻輳制御の仕組みをアプリケーションに組み入れるべきかもしれない。また、極端に遅い端末などは、別セッションで再送するほうが望ましい場合もある。このように、マルチキャストアプリケーションが要求を実現するためには、その程度に応じてトランスポート層からアプリケーション層まで様々な技術を要求の程度に応じて使い分けることが重要となる。このイメージ図を図6.11に示す。[18]

[18]図中のALF拡張については後述する。

図 6.11: 損失程度に応じた問題解決



第 7 章

ツールキットの提案

高信頼マルチキャストに関する研究の進捗度、前章で見てきたような具体的な問題点から、高信頼マルチキャストの実用化のためには次のような取り組みが必要と考える。

1 アプリケーション要求を体系的、定性的に整理する取り組み

2 具体的なアプリケーション事例の解を構築する中で、システムの種別ごとの設計指針を確立する取り組み

3 既存のプロトコルを容易に改善実装するためのプロトコル構築環境の整備

4 実装したプロトコル、アプリケーションを即座に実証実験するための環境の整備

その中で、本章では、特に2、3に着目して高信頼マルチキャスト構築ツールキット (M-RMT:Muramoto Reliable Multicast Toolkit) を提案する。MRMT は、その性格の差異から次の2つに分類される。

- MRMT A-kit
  高信頼マルチキャストアプリケーションの設計手法を具現化したソフトウエア群
- MRMT P-kit
  高信頼マルチキャストプロトコルを構築するためのソフトウエア部品群

以後それぞれについて説明する。

図 7.1: MRMT A-kit のイメージ

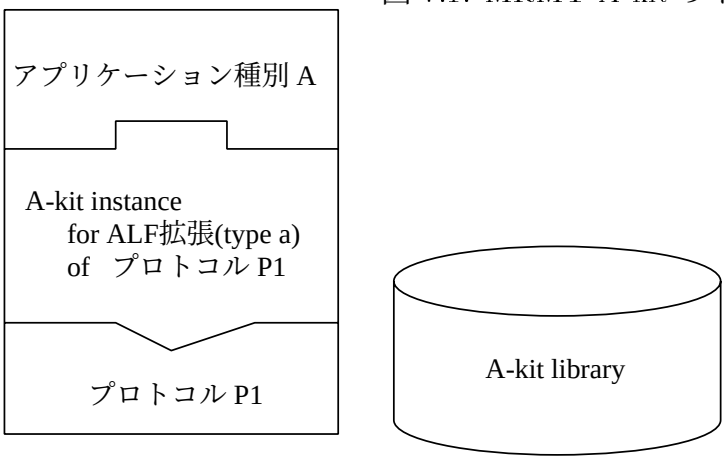
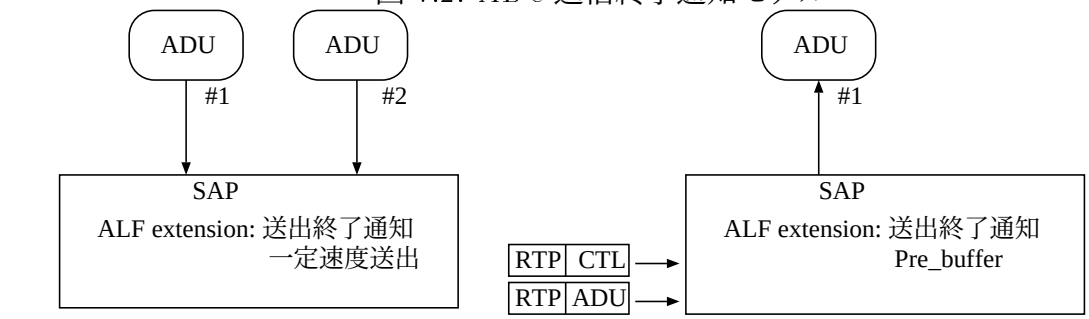


図 7.2: ADU 送信終了通知モデル



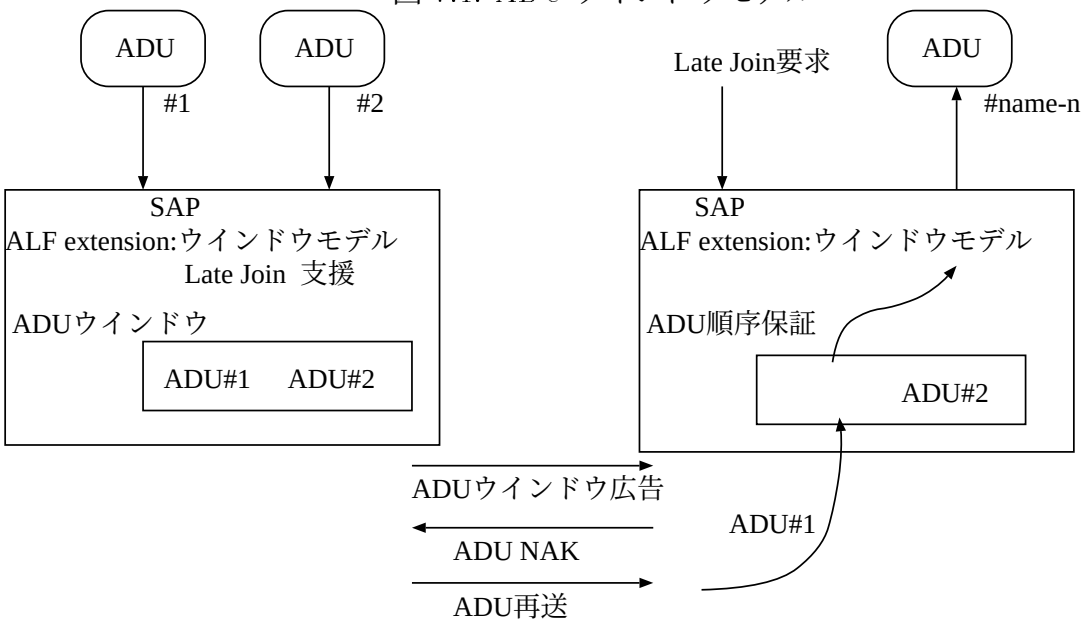
API :

setsockopt: Send_EOF ON 送出終了通知モード利用
setsockopt: Send_Rate rate 送出速度の指定
setsockopt: Set_ADU char * adu , int len ADUの指定
setsockopt: Start_Send(ブロック) 送出開始指示

API:

open ("rADU")
setsockopt: pre_buffer_size
read(soc) : return value = 0 EOF
select : wait for pre_buffer

図 7.4: ADU ウインドウモデル



このモデルは、具体的に libsrm[1] で実装されている。

ADU ウインドウモデル

マルチキャストアプリケーションには、ヒット曲 (MP3 形式) や、新聞のデータ、株式市況データなどの実時間性の高いデータの配送を定期的に配送するものが考えられる。これらのアプリケーションで配送課金される基本データ単位は、曲、記事、ある時点のある分野の市況などが考えられる。これらの情報は、配送時点での価値が最も高く、時間経過とともに比較的急速に価値が下がっていくものと考えられる。したがって、セッションにおいて配送されるべき ADU は、最新情報でありセッションへの途中参加者が請求する ADU も比較的最近に配送されたデータに集中することが予測される。

このような要求を容易に実現する枠組のとして図 7.4 で示すような枠組を提案する。送信者は、過去 n 回にわたって送信した ADU を送信ウインドウに保持しており、再送要求に対して高速な再送を保証する。

7.1.3 設計と実装

前章でしめしたアプリケーションを容易に構築するため ALF 拡張の ADU 送信終了通知モデルを実装した。送信者が送信を指示した ADU は、必要であれば適当なパケットサイ

[1] http://www-mash.CS.Berkeley.EDU/mash/software/srm2.0/

表 7.1: ALF 拡張 ADU 送信終了通知モデルの API

API	説明
送信者 API	
mrmt_tx_open	送信者 socket の open
mrmt_setsockopt	最大転送速度の指定
mrmt_write_real_adu_block	adu の書き込み、引数に adu とその長さとタイムスタンプ値を指定する
受信者 API	
mrmt_rx_open	受信者 socket の open (TSI を指定しないときは raw ソケット)
mrmt_tsiscan	最初のパケットの TSI を取得する (raw ソケット時のみ利用可)
mrmt_rx_init	共有メモリを確保し指定量の adu をプレバッファする。プレバッファ後、fork した受信プロセスが共有メモリに adu を受信し続ける。
mrmt_rx_read_adu	adu を読み出す。タイムスタンプ値も同時に読み出す。
mrmt_rx_term	受信プロセスの終了と共有メモリの開放を行なう。
共通 API	
mrmt_close	socket を閉じる

ズに断片化され RTP ヘッダとともに受信者に PGM で送付される。受信者のアプリケーションでは、プレバッファサイズを指定して socket を初期化したのち、ADU 単位のデータをタイムスタンプ値ともに読み込むことができる。

損失率が一定以下である状況で、送信者が固定長の ADU を一定レートで送信し、受信者が必要なプレバッファサイズを指定して socket を初期化すれば、受信者アプリケーションは連続ストリームの受信といった寛容な配送サービスを享受できる。

API

ADU 送信終了通知モデルを実現する API を、表 7.1に示す。

実装

送信者では、必要であれば ADU を経路の最小 MTU などに断片化して送付することができる。断片化されたデータは PGM を用いペイロードに RTP ヘッダを付与し送信者アプリケーションが指定するタイムスタンプを刻印して転送している。受信者では、フレー

表 7.2: ALF 拡張 API を適用しない時した時の MP3 配送プログラムの行数

プログラム名	ALF 拡張なし	ALF 拡張あり
mp3_send.c	256	213
mp3_recv.c	223	132
buf	107	0(必要なし)

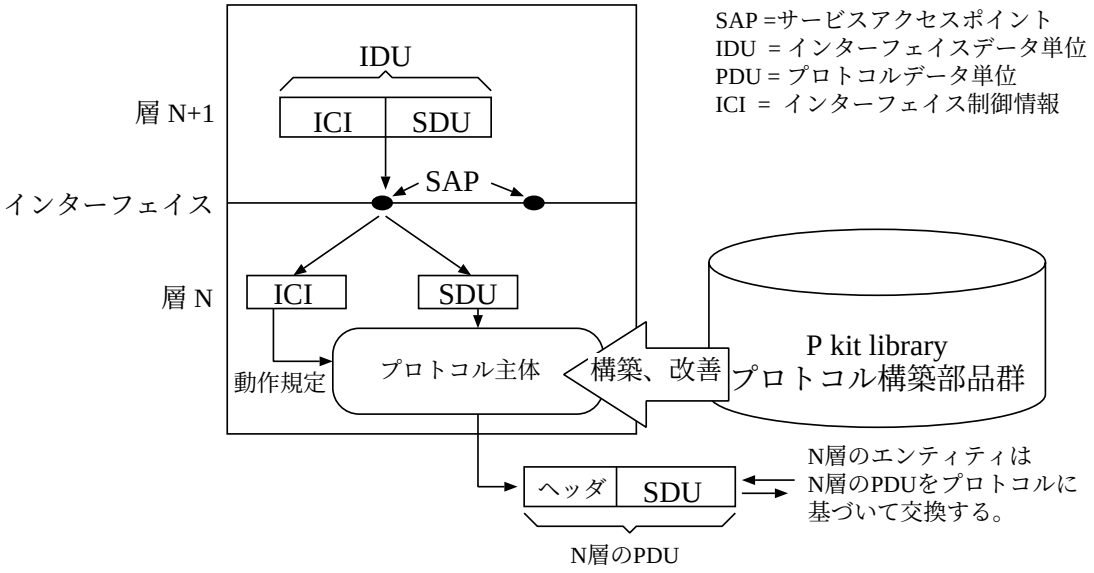
ム境界の判定にこれを用いると同時に、ADU のデマルチプレクス時にアプリケーションにタイムスタンプ値を通知する。受信者は、mrmt_rx_init 関数で、バッファサイズ、プレバッファサイズを指定できる。実装においては、共有メモリを用いた可変長のリングバッファを実現したので、遅い事が知られている select システムコールを使わずに、受信者アプリケーションを容易に実装することを可能となっている。

適用例

ADU 送信終了通知モデルを MP3 配送アプリケーションに適用する事により、実時間マルチキャストアプリケーションが容易に構築できることを確認した。[2]表 7.2 に送信者および受信者プログラムの行数を示す。socket の操作に関する記述、RTP ヘッダの扱いに関する記述、バッファリング処理に関する記述が簡略化できるため、アプリケーションのコード量が削減できていることがわかる。

[2]実装は、lss5-is26: emuramot/pgm-app/mp3-alf.tar にから入手可能である。

図 7.5: プロトコルスタックと P-kit の関係



7.2 P-kit

現在、高信頼マルチキャストプロトコルは標準化作業中である。アプリケーション種別ごとに最適なプロトコルを構築ブロックとして標準化されることが望ましい。このような状況では、プロトコルを即座に改定、実証実験できることが重要である。MRMT P-kit は、高信頼マルチキャストプロトコルを容易に改定するためのプロトコル部品群から構成されるツールキットである。

7.2.1 P-kit の特徴

ネットワークプロトコルは複雑さを減少させるために、階層化された設計実装が行なわれる。ある階層は一つ上位の階層に対してサービスを提供し、上位の階層は下位層のサービスを利用するのみでその実現の詳細には関知しない。プロトコルスタックと P-kit の関係を図 7.5に示す。上位層か階層へのサービスの要求は、サービスアクセスポイントを通して行なわれる。上位層と下位層間での情報の交換には、インターフェイスデータ単位 (IDU) を用いて行なわれ、データの伝達にはサービスデータ単位 (SDU) で行なう。各層でプロトコル主体 (entity) は、同一層のプロトコル主体とプロトコルデータ単位 (PDU) を交換する。その層のプロトコル主体はプロトコルデータ単位に適当なヘッダを付加してプロトコルに基づき送信あるいは受信相手の同一層のプロトコル主体と通信を進める。

P-kit は、このような動作を行なうプロトコル主体を容易に構築するための部品群である。

7.2.2 P-kit のアーキテクチャ

P-kit を用いて構築するプロトコル主体のアーキテクチャについて説明する。

状態遷移モデル

プロトコル主体は状態遷移機械に抽象することができる。その状態は次の 5 つで規定される [6]。

- 状態の有限集合
- 状態遷移規則の有限集合
- 述語の有限集合
- 入力事象の有限集合
- 出力事象の有限集合

状態遷移機械への入力は

- N+1 層のサービスから
- N-1 層のサービスから
- N 層のローカル処理 (タイマ) から

あると考える。

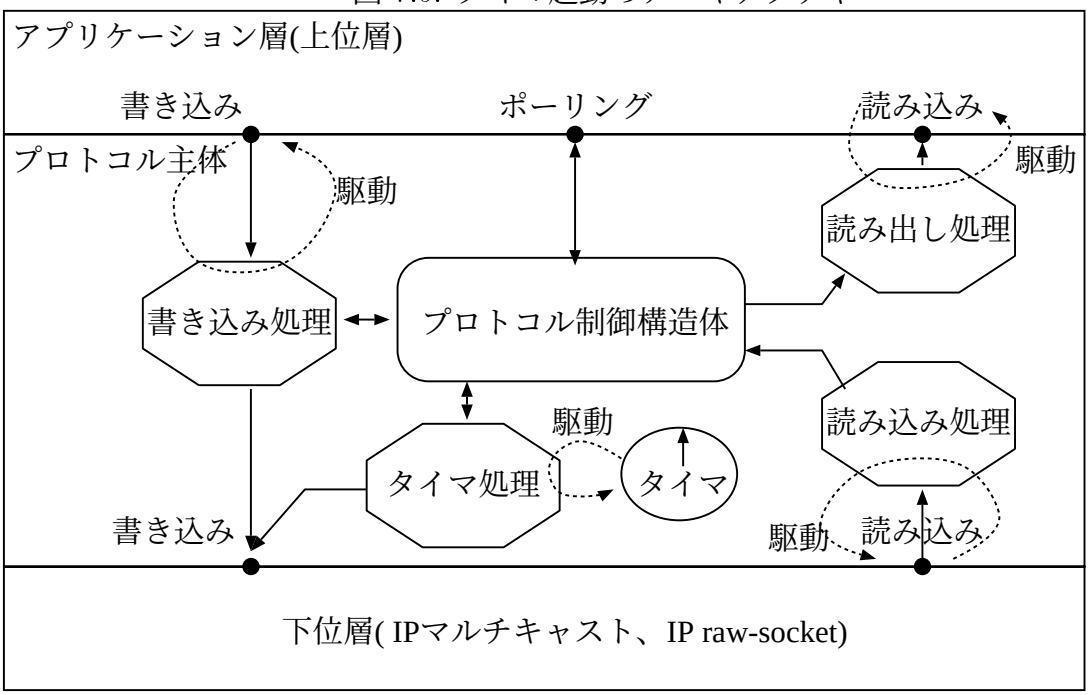
タイマ起動のアーキテクチャ

プロトコル主体は、アプリケーションの要求によって生成 (instanciation) され、アプリケーション層 (上位層) からの書き込み動作、プロトコル主体が持つタイマのイベント、下位層からのパケットの到着によって駆動される。この様子を図 7.6 に示す。プロトコル制御構造体はプロトコル主体の状態、SDU のバッファ、イベントの待ち行列を要素保持している。上位層、下位層、ローカルタイマからのイベントは、プロトコル制御構造体の状態を変化させながら一連の通信を進行させる。

7.2.3 プロトコルの基本機能

P-kit は、プロトコル主体を駆動するアーキテクチャの雛型とプロトコルを構成する部品群とから構成される。表 7.3 に P-kit を構成する部品群の候補を示す。

図 7.6: タイマ起動のアーキテクチャ



7.2.4 設計

P-kit を用いて様々な高信頼マルチキャストの実装を行なう事ができる。具体的にPGM プロトコルの終点ホスト実装の設計を次にしめす。

PGM を実現するため設計

図7.7はタイマー起動のアーキテクチャを用いたPGM 終点ホストの設計を示している。

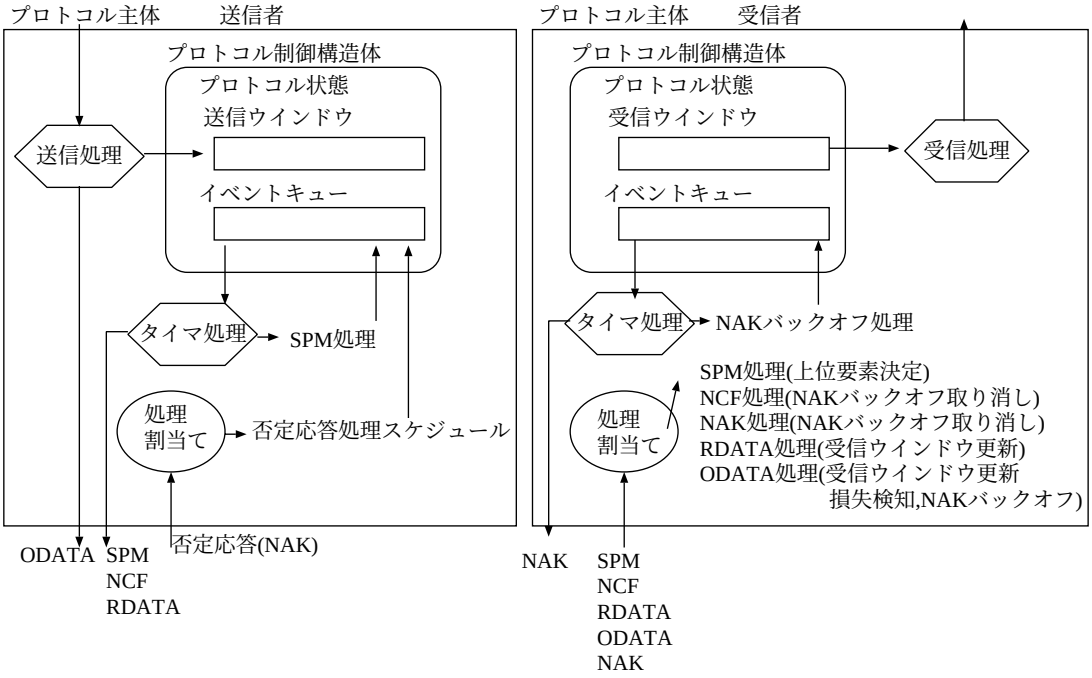
送信者の送出処理は、プロトコル制御構造体の送信ウインドウにSDU を格納した後、PDU(この場合ODATA) を送付する。送信者が受けとったNAK 処理はイベントキューにスケジュールされ、タイマにより起動される。

受信者が受けとったPDU はその型によって該当する処理が割り当てられ実行される。NAK の送出はイベントキューにスケジュールされる。該当するRDATA の受信でスケジュールは取り消される。

このPGM 実装で用いられるP-kit 部品は次の通りである。

- Byte ordering
- Checksum

図 7.7: P-kit を用いたPGM 終点ホスト設計



- Inactivety control
- Sequence number
- Negative acknowledgement
- Segmentation
- Rate flow control
- Timer

表 7.3: P-kit の部品群

部品名	意味
Blocking	複数の SDU を一つの PDU として送出する
Byte ordering	バイトオーダー変換
Checksumming	チェックサム処理
Concatenation	複数 PDU から一つの SDU を構成する
Connection control	接続管理を行なう
Encapsulation	カプセル化処理
Encryption	暗号化
FEC	前方誤り訂正処理
Inactivity control	通信相手へのポーリング
Jitter compensation	ジッタ圧縮処理 (バッファリング、ペイシング)
Rate flow control	定レート送出、レート変更処理
Segmentation(flagment)	一つの SDU を複数の PDU に断片化する
Traffic shaping	シェイピング (バッファリング、ペイシング)
Policing	一定レート以上のパケットを削除
Window flow control	ウインドウ広告にもとづく送信量制御
エラー制御	
Sequence number	通番付与、管理
Acknowledgement	肯定応答処理
Negative acknowledgement	否定応答処理
Retransmission	再送処理
Timer	イベントキューとタイマ処理

第 8 章

今後の課題

高信頼マルチキャストに関する研究は、本格的普及展開のための研究に比重が移行されていくものと考える。高信頼マルチキャストの構築ブロックは、既に提案されているプロトコルの特性を明確にし、アプリケーション種別ごとに最適な組合せとシステムが正常に動作するためのネットワークトラフィック特性、保証されるべきサービス特性が明確になる形で整備されていく必要がある。したがって、今後次のにあげるような取り組みを行ない研究を加速させる必要がある。

8.1 実証実験の実施とプロトコル拡充

高信頼マルチキャストの動作と一般トラフィックとの親和性を評価する。現実的なネットワーク資源の消費量の範囲内で、実用的なアプリケーション品質が得られためのプロトコル改善要求の抽出や、効率的にアプリケーション構築を行なうための技術蓄積を行なう。

8.1.1 一般トラフィックによる損失発生状況での適合型アプリケーションの実証実験

レートベースで送信を行なうマルチキャスト通信のインターネット上でどのような損失し、それを吸収するトランスポートプロトコル、適合型アプリケーションはどのように実現されるべきかを体系的に調査することは有効である。具体的には適当なアプリケーションを選択し、次にあげるようなネットワーク環境化での実証実験があげられる。

- 統合サービスアーキテクチャが提唱する制御負荷サービス配下での実証実験

- 衛星回線の降雨状態における損失状況での実証実験

8.2 ツールキットの拡張

様々な、実証実験を容易に行なうために、本研究で提案したようなツールキットを開発拡充し、プロトコルおよびアプリケーションを構築改定、即座に遠隔配置できるようにすることは重要である。
ここでは、本研究で提案したツールキットを発展させる方向性についてのべる。

8.2.1 高信頼マルチキャストの要求記述記述とプログラム記述からトラフィック要求記述 (Tspec)、送信者、受信者プログラムを生成する枠組の考察と設計実装

高信頼マルチキャストアプリケーションを容易に生成するRPCのようなツールキットを設計開発することは有効であると考える。

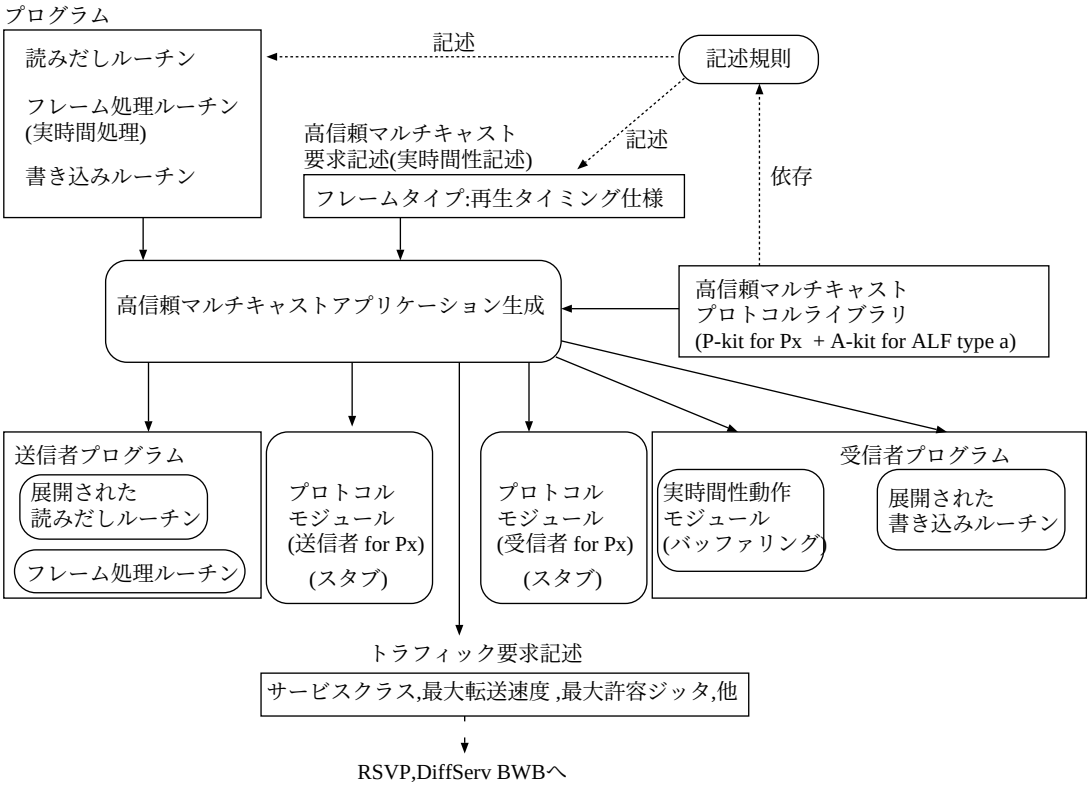
着目するプログラム構造について説明する。マルチキャストアプリケーションは送信者、受信者、必要なら中継者アプリケーションから構成される。送信者は、配送すべきデータ、ストリームを特定の装置、もしくはファイルなどから読み込み、指定した速度でネットワークに送出する。受信者アプリケーションは、読み込んだデータを必要であればバッファリングしてアプリケーションの動作を特徴づける処理へデータを引き渡す。

例えば、本実験で取り上げたDVTSでは、送信者はIEEE1394装置から一連のストリームデータを読み込み、フレーム境界の判定を行なった後、必要なストリームを送出する。受信者は、受け取ったストリームを特定時間バッファリングして連続した再生動作を行なう。

このようなプログラム構造に着目して、送信者、受信者アプリケーションを生成するツールキットを考案することができる。そのイメージを図8.1にしめす。

高信頼マルチキャストの要求 (例えば実時間性に関する要求) の記述と、一定規則に則って記述された送信者および受信者の動作双方を記述した一本のプログラムから、送信者および受信者のアプリケーションを生成することができれば、アプリケーション動作のテストを遠隔配置するまえにテストすることが可能となる。予め動作保証されたアプリケーションは、高信頼マルチキャスト配送がサポートされているネットワークへ展開しても正常に動作することが保証される。また、利用する高信頼マルチキャストプロトコルを変更する作業についても容易にできることが期待できる。

図 8.1: RPC のような高信頼マルチキャストツールキット



8.3 アプリケーション設計実装によるプロトコルへの要求の整理

マルチキャストアプリケーションには、要求する特性の異なる複数のマルチキャストセッションで構成させることが考えられる。マルチキャスト特有のアプリケーション要求である同時性といった要求や、鍵配送、サービス否定攻撃に対する対抗手段の確立などアプリケーション要求を満たすために、高信頼マルチキャスト配送網が満たすべき性質を体系的に抽出することは重要である。具体的なアプリケーションに着目して要求を整理する作業をあげる。

8.3.1 高信頼マルチキャストプロトコルを用いたリアルタイムオークションのためのシステム設計とプロトコル要求整理

インターネットの普及は、商品の販売流通形態に大きな変革をもたらしたと言われている。また、消費者主体のまとめ買い行動による価格交渉や、逆オクションに代表されるよ

うに、もの、サービスの価格決定構造の変革も起こりつつある。

高信頼マルチキャストを用いて公平に同時に情報を受信者に提供できれば、実際のオクションに近い環境がネットワーク上に構築できる。競り落す楽しみや、場の持つ雰囲気による興奮など今までのアプリケーションでは提供できなかった新たな価値を提供できる可能性がある。

このようなアプリケーションでは配送されるデータの同時性が、重視される。複数の送信元から送付されるパケットの順序保証が必要となるかもしれない。場への参加費用を徴収する場合や、プライバシーに関わる商品を扱う場合などセッション全体を参加者以外から守るため、暗号化がなされるべきかもしれない。競り落した商品の取引が確かに行なわれたという情報はマルチキャストされるべきかもしれない。

想定する受信者の規模とネットワークの直径、1回のオークションにかかる平均時間、それを実現するための鍵配送のレイテンシ、など設計を詳細に進めるに際してプロトコルへの要求が浮き彫りになることが期待できる。

8.3.2 音楽著作者が自ら著作物を発表同時販売するためのシステム設計と要求整理

多様化した個性が重視される時代に、画一的な情報の配信は不要かもしれない、ネットワーク上に配置された情報やコンテンツを要求に応じて提供できる仕組みのようがより重要視されるべきかもしれない。しかし、一方で洗練された情報やコンテンツを誰よりも先に手にしたいという欲求はなくならない。すぐれた著作物を提供する作者は、あるものは無償で配り、利用者との交流を深めるための共通語を形成するために用い、使い洗練されたものを有償で発表と同時に多数販売したいと考えるかも知れない。

このような場と提供するアプリケーションは、著作物を発売と同時に配信する仕組み、配信が完了できた相手先から確実に集金するための枠組が必要となる。受信者の数は数百万以上を想定する必要があるかもしれない。想定する受信者の能力のバラツキ、コンテンツ転送に許される最大時間、課金のための情報収集のために許される最大時間、フィードバックの爆発を防ぐために最適なプロトコルの選定、中継者に必要な資源と設置場所、マルチキャストとFirewallとの親和性[1]検討することは実用化のための第一歩となる。

[1]利用者は、防火壁(Firewall)の向こう側にいるかもしれない

第 9 章

まとめ

本論文では、高信頼マルチキャストに関連の深い技術の標準化の動向を紹介した。その中で、多彩なアプリケーション要求を定量的にとらえる取り組みや、構築ブロック方式による高信頼マルチキャストプロトコルの標準化方法、通信の品質を規定する統合サービスアーキテクチャに関する動向を概観した。

具体的なアプリケーションの設計実装を行なう中で、顕在化する問題点を指摘し、高信頼マルチキャストの研究を推進するためのツールキットを提案した。

ツールキットは、その性質の差異からアプリケーションの構築を容易に行なうA-kit、プロトコルの改善拡充のためのP-kitに分けて設計、一部実装し音楽配信アプリケーションに適用、その有効性を確認した。

最後に、ツールキットの発展計画を含めて高信頼マルチキャストの研究を加速するための取り組みついて言及した

謝辞

本研究を進めるにあたり指導教官である篠田陽一助教授には、機を見てその折に適切な助言を多数頂いた。そして、WIDEプロジェクト、落水、篠田研究室の方々からも様々な有益な意見を頂いた。また、松下電器産業株式会社には、このような研究を行なう貴重な機会を与えていただいた。記して、ここに感謝の意を表したい。

参考文献

[1] W. リチャード・スティーヴンス 著,篠田 陽一 訳, UNIX ネットワークプログラミング,トッパン, 1992.

[2] W. リチャード・スティーヴンス 著,篠田 陽一 訳, UNIX ネットワークプログラミング 第2版 Vol.1,トッパン, 1999.

[3] W. リチャード・スティーヴンス 著,井上 尚司 監訳, 詳解 TCP/IP,ソフトバンク, 1997.

[4] Andrew S.Tanenbaum 著, Computer Networks 3rd-edition,Prentice Hall PTR, 1996.

[5] Miller,C.Kenneth 著, Multicast networking and applications,Addison Wesley Longman,Inc, 1999.

[6] Stefan Boecking著, Object-Oriented Network Protocols,Addison-Wesley, 1999.

[7] Dave Kosiur著,苅田 幸雄 監訳マスタリングTCP/IP IPマルチキャスト偏,オーム社, 1999.

[8] B. Braden, USC/ISI, D. Clark, MIT LCS, J. Crowcroft, UCL, B. Davie, Cisco Systems, S. Deering, Cisco Systems, D. Estrin, USC , S. Floyd, LBNL, V. Jacobson, LBNL, G. Minshall, Fiberlane, C. Partridge, BBN, L. Peterson, University of Arizona, K. Ramakrishnan, ATT Labs Research, S. Shenker, Xerox PARC, J. Wroclawski, MIT LCS, L. Zhang, UCLA, " Recommendations on Queue Management and Congestion Avoidance in the Internet" IETF RFC2309, April, 1998.

[9] A. Mankin,USC/ISI , A. Romanow,MCI ,S. Bradner, Harvard University, V. Paxson ,LBL , "IETF Criteria for Evaluating Reliable Multicast Transport and Application Protocols" IETF RFC2357, June, 1998.

[10] P. Bagnall, R. Briscoe, A. Poppitt,BT, " Taxonomy of Communication Requirements for Large-scale Multicast Applications" IETF RFC2729, Dec, 1999.

[11] B. Whetten, Talarian, L. Vicisano, Cisco,R.Kermode, Motorola, M.Handley, ACIRI, S.Floyd, ACIRI, " Reliable Multicast Transport Building Blocks for One-to-Many Bulk-Data Transfer" IETF Internet draft June, 1999

[12] M. Handley, ACIRI,B. Whetten, Talarian,R. Kermode, Motorola, S. Floyd, ACIRI,L. Vicisano, Cisco "Reliable Multicast Design Space for Bulk Data Transfer" IETF Internet draft June, 1999

[13] Brad Cain,Nortel,Tony Speakman,cisco,Don Towsley,UMASS "Generic Router Assist (GRA) Building Block Motivation and Architecture" IETF Internet draft Oct, 1999

[14] 山内 長承,城下 輝治,佐野 哲央,塩川鎮雄,日本IBM東京基礎研究所, NTT情報通信研究所, "高信頼同報バルク転送プロトコルRMTPとReliable Multicastの研究動向" http://www.trl.ibm.co.jp/projects/rmtp/wide_ps.gz

[15] Sally Floyd,Van Jacobson, "Random early detection gateways for congestion avoidance" IEEE/ACM Transaction on Networking,Vol.1,No.4,pp.397-413, August 1993

[16] 山内 長承,佐野 哲央,城下 輝治,高橋 修,日本IBM東京基礎研究所, NTT情報通信研究所, "高信頼マルチキャストにおけるフロー・輻輳制御" http://www.trl.ibm.co.jp/projects/rmtp/icf_ps.gz

[17] Tony Speakman ,Nidhi Bhaskar , Richard Edmonstone, Dino Farinacci, Steven Lin , Alex Tweedly , Lorenzo Vicisano ,Cisco, "Reliable Multicast Transport Building Blocks" IETF Internet draft June, 1999

[18] Sally Floyd,Van Jacobson,Steven McCanne,Lawrence Berkeley Laboratory, "A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing" http://www-nrg.ee.lbl.gov/floyd/srm-paper.html Proceedings of SIGCOMM '95 (Boston, MA, Sept. 1995), ACM.

[19] Luigi Rizzo, "Effective Erasure Codes for Reliable Computer Communication Protocols" http://www.iet.unipi.it/ luigi/fec_ccr.ps.gz

[20] Clark, D. D., and Tennenhouse, D. L. "Architectural Considerations for a New Generation of Protocols" Proceedings of SIGCOMM '90 (Philadelphia, PA, Sept. 1990), ACM.

[21] Luigi Rizzo,Lorenzo Vicisano, J.Crowcroft "The RLC multicast congestion control algorithm" http://www.iet.unipi.it/ luigi/rlc99.ps.gz

[22] " Specifications Consumer-Use Digital VCR's using 6.3mm magnetic tape", HD Digital VCR Conference, 1994 society, 1995

[23] "IP レベルの誤り訂正プロトコル(IP-FEC)", WIDE プロジェクト研究報告書 1998 年度版, P350〜P365