

第 XI 部

信頼性を有するマルチキャスト通信 技術

第11部

信頼性を有するマルチキャスト通信技術

第1章 はじめに： WIDE での Reliable Multicast 研究の進展

1.1 環境

インターネットの大容量化、放送型サービスやTV会議などの応用への関心が高まる中で、マルチキャストの安定的な利用への要求が顕在化している。具体的には、一方ではマルチキャストの経路制御技術の安定化の要求があり、他方では1対1通信でTCPが実現しているような広範な信頼性への要求がある。

WIDEの研究範疇でも、JGN/JBを利用したマルチキャスト実験が頻繁に行われ、SOI やさまざまなイベントにおいて、デジタルビデオ (DV) 動画のマルチキャスト転送が行われた。その結果として求められたのは、安定なサービスであり、それを実現する技術として経路制御の安定化とトランスポート層における Reliable Multicast (RM) 機能の実現であった。

US における Reliable Multicast 研究も、IETF RMT-WG での標準化活動が標準を策定するという意味では大きな進展は見られなかったが、新しい技術の提案、要素技術間の関連づけなどにおいて、議論の進展を促していたように見られる。一方で、昨年度以前に企業から提案されていたたとえば PGM のような技術が爆発的に開花することもなかったが、最近 Digital Fountain 社による、実用性を意識した FEC に関連する研究、具体的には Tornado Code や Ruby Code の開発は目を引いた。

1.2 活動

WIDE における 2000 年度の RM 研究グループの活動は、参加者の所属研究室における個別の研究と、WIDE 研究会や合宿などにおいてお互いに成果を検討し合う機会とから成っていた。研究の具体的な内容は次のように整理できる。

- FEC と PGM による Reliable Multicast の実

装と評価

- 動的なプロトコルの選択を用いた高信頼性マルチキャスト通信の評価と応用
- JGN/JB 環境での DV 転送における FEC の評価
- FEC の実ネットワーク環境における回復性能とモデル化

前者の2件は、RM 実現のために提案されているさまざまな方法をうまく組み合わせることにより、安定なトランスポートを実現しようとする研究といえるだろう。後者の2件は RM の基礎技術の1つである FEC の性能に関して、実トラフィック環境での評価やそれを踏まえたモデル化を試みたものである。

2001 年 3 月の合宿の BOF では、共通の認識として

- 全体としてみると
 - － 要素技術 (SRM や PGM のような方式、FEC の技術)
 - － この要素技術を組み合わせるやり方や効果に関しての研究は、かなり進んだといえるだろう。ある程度評価を下せるようになった感がある。
- 今後は、実環境においてこれらの評価が正しいか、考えていない問題として何があるか、などを見極めるために、実環境において、実アプリケーションを支えるような、RM の実用実験を考える必要がある。

などが挙げられた。今後の1つの研究指針となるものとする。

本報告書では、上に記した個別の研究のうち、前者の2件を紹介する。

第 2 章 FEC と PGM による Reliable Multicast の実装と評価

2.1 はじめに

インターネットのトラフィック量および接続ノード数は急激に増加し、エンドエンド間は地球規模に拡大している。また、新しいアプリケーションの登場により、従来の Point-to-Point 型の通信形態から、エンドエンドでの大規模なマルチキャストサービスの提供が必要とされてきている。

しかし、現在の IP マルチキャスト通信は UDP に基づいており、信頼性のあるデータ通信は提供していない。また、マルチキャスト通信を必要とするアプリケーションが要求する信頼性は様々であり、研究分野では、基本的なデータ通信の信頼性保証機能だけでなく、フロー制御、輻輳制御、FEC、セキュリティ、グループメンバシップなどのより高度な機能の研究や、受信ノード数が不均一な環境への適用などが進められ、多数の Reliable Multicast (高信頼性マルチキャスト通信) が提案されている。

IETF の RMT (Reliable Multicast Transport) WG では、Reliable Multicast で最も利用ケースが多いと思われる one-to-many 型のバルクデータ転送に絞って標準化を進めている。バルクデータ転送とは、輻輳制御が可能ないようにデータストリームが連続的に長時間続く転送と定義されており、大容量ファイルだけでなく、音声・動画ストリームなども対象としている。また、標準化にあたっての要求条件は、TCP friendly な輻輳制御、受信ノード数の 1000 ~ 10000 のスケーラビリティ、セキュリティなどが挙げられている [104]。

文献 [104] で示されているように、マルチキャスト通信に対する要求はアプリケーション毎に様々であり、単一のプロトコルで Reliable Multicast を設計することは不可能である。そのため、標準化アプローチとしては、Building Blocks 方式が検討されている。Building Blocks 方式とは、プロトコルの様々な機能単体 (Protocol Component) を識別し、それらを組み合わせることによって、特定のアプリケーションに適応したプロトコルを構築する方式である。Building Blocks 方式による最終的なプロト

コル (Protocol Instantiation) も 1 つに絞るのではなく、現在、以下に示す 3 つのプロトコルの標準化を進めている。

- NACK-based protocol
- Tree-based ACK protocol
- Asynchronous Layered coding protocol

RMT WG の Building Blocks 方式による標準化は始まったばかりであり、単一のプロトコルではなく 3 つの Protocol Instantiation の同時標準化や、共通 Building Block の部品化など、これまでになく複雑な標準化方法を取り入れている。そのため、Building Blocks 方式の Reliable Multicast に対する有効性を評価することは重要である。

本章では、Building Block の構成機能である、Open Loop 型パケット損失回復プロトコルとして IP-FEC on IPv4 (Co. TOSHIBA corp.) と、NACK 型パケット損失回復プロトコルとして PGM on IPv4 (Co. L. Rizzo) を、IPv6 上で実装した。さらに、これらを組み合わせて、IPv6 マルチキャストプロトコルである PIM-SM (Protocol Independent Multicast-Sparse Mode) に適用することで、受信ノード数に対してスケーラビリティを保証した Reliable Multicast を実装し、評価する。実測による評価から問題点を明確化し、考察を与え課題を抽出する。

2.2 FEC (Forward Error Correction)

2.2.1 FEC 概要

FEC はエンドエンド間の通信で用いることを基本としている。使用形態は図 2.1 となる。

送信ノードは実データ (図 2.1 では data) の他に、誤り訂正用コードを (図 2.1 では code) として送信する。IPv6 フラグメント機能の拡張として実装しており、実データパケットも誤り訂正用コードパケットも、途中ルータでは IPv6 フラグメントパケットとして転送される。したがって、転送途中ルータでは、誤り訂正コードの生成/復元の演算処理を行わない。

2.2.2 提案方式

本章では、IP-FEC on IPv4 (Co. TOSHIBA corp. [177]) を KAME (IPv6) に移植し、Layer3 の FEC を実装している [195]。IPv6 フラグメントの拡張機能として実装しており、IPv6 フラグメントヘッダの

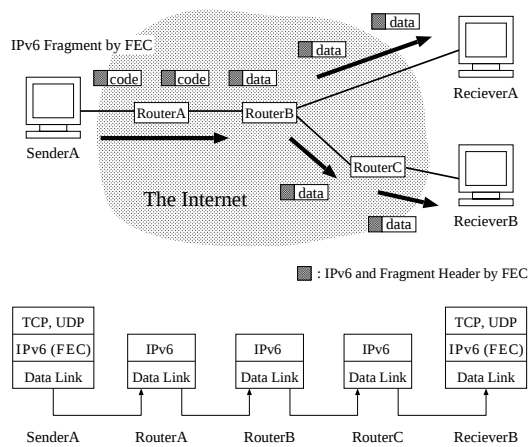


図 2.1. FEC の使用形態

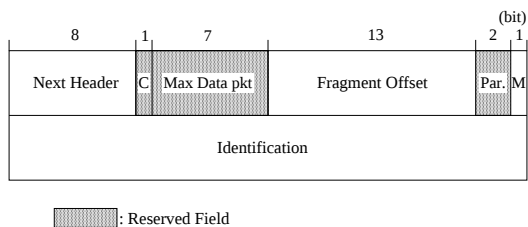


図 2.2. IPv6 Fragment Header Format by FEC

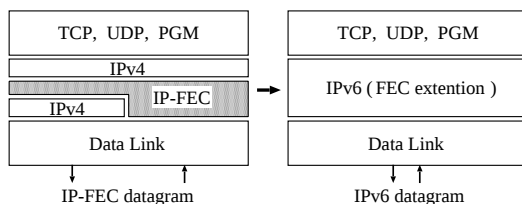


図 2.3. 実装的 Layer 構造

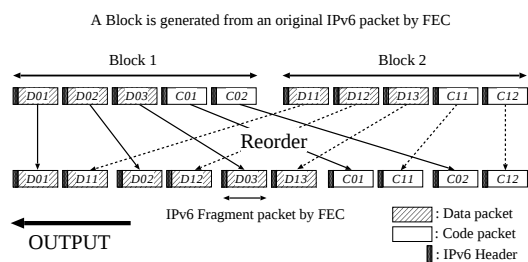


図 2.4. インターリーブオプション

Reserved フィールドを使用する (図 2.2)。

IP-FEC on IPv4 からの改善点として、IP-FEC on IPv4 独自のカプセル化による IP 層の重複処理によるオーバーヘッドを軽減し、FEC 未実装ノードとのデータ通信を可能としている (図 2.3)。

また、オプション機能として、パケットの送信パターンを図 2.4 のように再配置 (Reorder) するインターリーブ技術を実装しており、数個連続してパケットを損失する短いバースト誤りに対して訂正能力を維持することができる。

FEC on IPv6 の利点

FEC は本来、トランスポート層で実装すべき技術であるが、Layer3 で実装したことにより、下位層データリンク技術に依存せず任意のデータリンクを経由するエンドエンドの通信に対して FEC を適用できること、トランスポート層に非依存とすることにより、多くのマルチキャストプロトコルとの組合せが可能になるだけでなく、多くのアプリケーションに対しても適用可能になることである。また、IPv6 は豊富なアドレス空間をもつため、受信者の多い大規模なネットワークで、エンドエンドで信頼性のあるデータ通信を提供できる。さらに、IP 層 FEC は pro-active であり、Reliable Multicast におけるスケラビリティを保証することができる。

パケット分割による FEC

本章での FEC は、IPv6 パケットを分割して誤り訂正コードを付加する手法を用いている。このとき、IPv6 フラグメント機能と FEC の分割機能を統一し、FEC を IPv6 フラグメントの拡張機能として実装した。これは、類似した機能を統一することで、冗長な処理の軽減を目的としている。分割手法の利点としては、細かいデータを取り扱うことにより、到着するパケット間のジッタを小さくすることでストリームデータの再生品質を高く維持できることが挙げられる。

Reed Solomon 符号

誤り訂正符号には Reed Solomon 符号 ($G(x) = x^8 + x^4 + x^3 + x^2 + 1$) を用いている¹。RMT WG では、他に Tornade 符号、LT 符号が提案されており [102]、アプリケーションの特性に応じて、選択的に符号系の変更を可能とすることが推奨されている。Reed Solomon 符号の特徴としては、処理速度が早い、生成できる誤り訂正符号数がスケールしにくい、という特徴がある。また、データ部をエンコードす

¹ Reed-Solomon 符号は 1 バイトを 1 シンボルとする誤り訂正符号である。

ることなく誤り訂正符号を生成するため、パケット損失がない場合、FEC 未実装ノードでもデータの受信が可能である。

2.3 FEC 性能評価

2.3.1 評価項目

FEC の評価項目として以下の 3 つを挙げる。

1. 処理速度・スループット

FEC は誤り訂正コードの生成・復元処理が高負荷であるため、FEC の処理が限界に達した場合、結果として通信性能の低下する。また、他のマルチキャストプロトコル・アプリケーションとの併用を考慮すると、十分なスループット・処理速度を保証する必要がある。FEC の演算処理速度をカーネル内で測定し、スループットの理論値を得る。さらに、無償で公開されている通信性能評価ツールである netperf[74] を用いて実際のスループットを測定する。

2. 損失パケットの復元率

受信ノードの多いマルチキャスト通信では、パケット損失に対する再送オーバーヘッドがネットワークの輻輳を引き起こす原因となる。そのため、実ネットワークにおけるパケットの損失パターンをパルス、バーストの 2 種類に分類し復元率を測定することで、FEC のパケット損失に対する有効性を評価する。さらに、FEC は誤り訂正能力以上のバースト的なパケット損失に対して効果が低い [177]、送信ノードにおいてパケット送信パターンを Reorder するインターリーブ技術を用いることで、誤り訂正能力を維持する。

3. 実データの品質

パケット分割方式の FEC は、ジッタを小さくすることにより高いデータ品質を保証できる。しかし、FEC は高負荷な演算処理を必要とするため、結果的に、ジッタが大きくなる可能性がある。そのため、実トラフィックとして、処理負荷の高い大容量ストリームである DV (Digital Video) を転送し評価する。

送信ノード、受信ノードの計算機環境を表 2.1 に示す。

表 2.1. 使用した主なソフトウェアとそのバージョン

OS	FreeBSD 3.3-RELEASE
	KAME 19991108-snap
CPU	PentiumIII 500MHz
Memory	256MB
NIC	Ethernet 100Mbps

2.3.2 処理速度・スループットの測定

実験環境として他の機器の影響が出ないように

- CPU 時間、メモリ使用量などを無制限にする
- 余計なプロセスを実行しない
- 2 台の計算機をクロスケーブルで接続

とした。これにより、FEC を使用しない場合の限界性能を推定できる。

カーネル内での処理速度の測定

処理時間の測定には、カーネル内で提供されている nanotime 関数を用いた。CPU は PentiumIII 500MHz である。測定結果を図 2.5 に示す。図 2.5 より、処理速度の理論値が得られた (表 2.2)。

表 2.2. パルス (loss 率 5%)

訂正コード	1	2	3	4
処理速度 (Mbps)	85.0	66.5	49.9	42.9

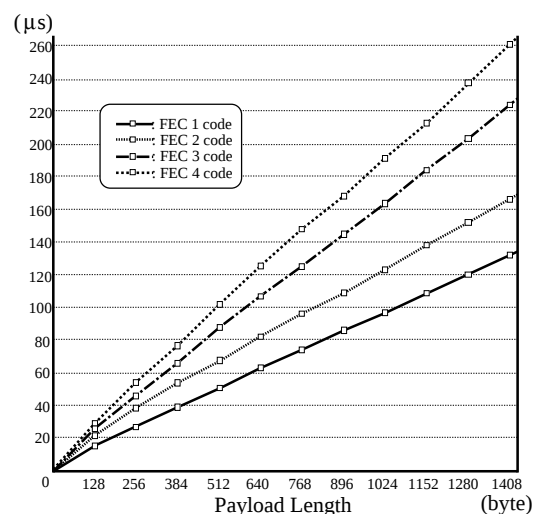


図 2.5. 処理速度のカーネル内での測定

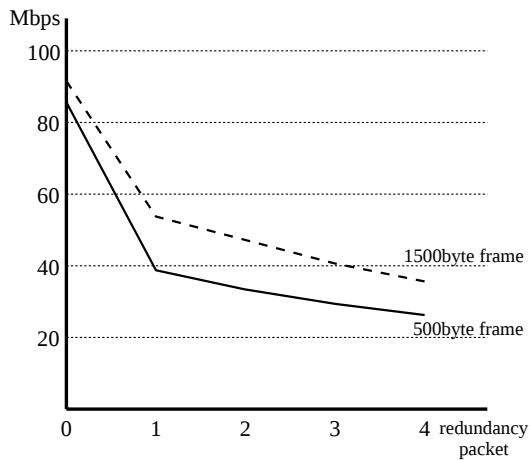


図 2.6. netperf によるスループットの計測

netperf による測定

IPv6 対応に変更した netperf2.1pl3 を用いて、FEC を使用した場合、使用しない場合のスループットを測定した (図 2.6)。IPv6 パケットが 500byte と 1500byte の場合について測定し、各々の IPv6 フラグメントのペイロード長を 104byte、304byte とし、実データパケットを 5 個にフラグメントした。誤り訂正コードは 1~4 個付与した。図 2.6 で誤り訂正コード数が 0 のときは IPv6 フラグメントを行わず、FEC を使用しない場合のスループットを示す。

500byte の転送において、FEC を使用しない場合、約 85Mbps 程度のスループットが得られる。FEC を使用した場合、26~38Mbps のスループットが得られ、約 30~45% に低下する結果となった。1500byte の転送において、スループットの低下率は若干減少した。

2.3.3 復元率の測定

パケット損失パターンとしてパルス、バーストを発生する機能を PC ルータのカーネル内に実装した。復元率は、このルータを経由したプライベートネットワークで測定した。

パルスのパケットロスに対する復元率

パルスのパケット損失をランダムに 1~10% 発生させた結果を図 2.7 に示す。分割データ数は 5 とした。

図 2.7 より、FEC はパケットのパルスの損失に対して、ほぼ完全に復元することができ、有効な誤り訂正制御方式であるといえる。

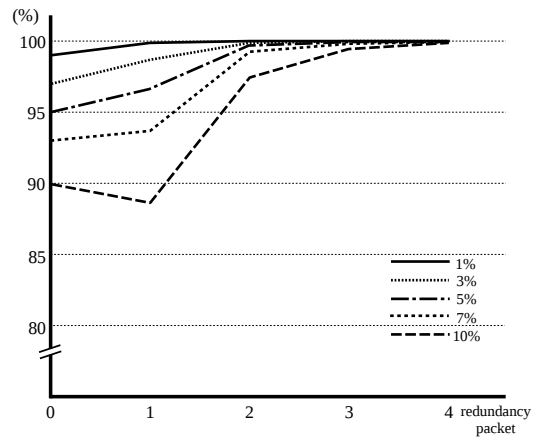


図 2.7. 復元率 (パルス)

バースト的なパケットロスに対する復元率

パケットを 5 個連続して損失するバースト誤りをトータルのパケット損失率 5% で発生させ、インターリーブオプションを使用したときの復元率を図 2.8 に示す。

図 2.7 より、FEC は、短いバースト誤りに対しても、インターリーブ技術により復元率を高く維持することが可能である。

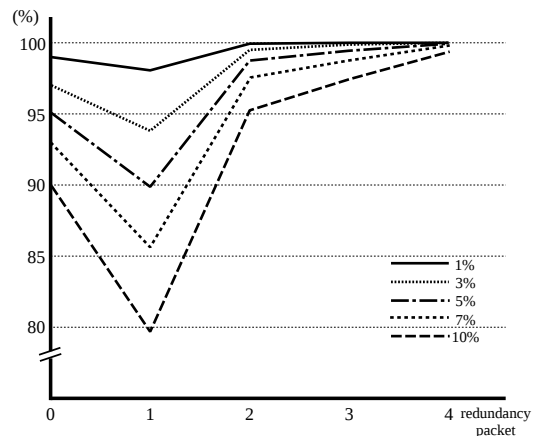


図 2.8. 復元率 (バースト、インターリーブ使用)

2.3.4 DV 転送によるデータ品質評価

実験環境を図 2.9 に示す。DV 機器を接続するために、IEEE1394 インターフェースを取り付けた送信・受信ノードを準備した。送信ノードには DV カメラを、受信ノードには DV デッキを接続し、DV カメラで撮影されている映像を受信ノードに送信した。受信ノードでは、受信した映像を DV デッキを介しモニタに出力する。実験に利用した FreeBSD には

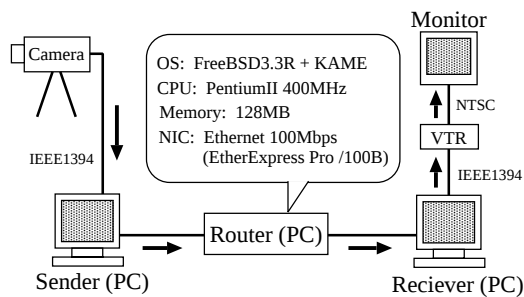


図 2.9. DV 転送実験環境

IEEE1394 のドライバを適用し、DV の送受信アプリケーションとして DVTS[1] パッケージを使用した。

本実験では、転送する DV は、DVTS のフレームレート制御機構により、 $\frac{1}{2}$ フレームとした (16Mbps)。これは、4.5.1 節のスループット測定結果より、FEC の 500byte のフレーム転送最大スループットが 26 ~ 38 Mbps であることから、FEC が訂正コードを付与した場合の DV のフルレート (32 + 6.4 ~ 25.6Mbps) の送受信処理ができないためである。

実験結果

- パルスの損失をランダムに 5% 発生させた環境での結果を表 2.3 に示す。FEC を使用しない場合、ブロックノイズが多く表示され、音声も途切れが目立ち、聞きとりにくい。しかし、FEC を使用することにより、画質・音声ともパケットロスがない品質と同程度まで改善した。ただし、誤り訂正コードを 4 個付与した

表 2.3. パルス (loss 率 5%)

訂正コード	画質ノイズ	音声
1	多	聞きとれず
2	少	途切れがなくなる
3	なし	良好
4	多	聞きとれず

表 2.4. パースト (loss 率 5%、インターリーブ使用)

訂正コード	画質ノイズ	音声
1	多	聞きとれず
2	少	途切れがなくなる
3	少	途切れがなくなる
4	多	聞きとれず

場合は、画質・音声とも急激に劣化した。

- パースト的なパケットの損失をトータルで 5% 発生させた環境で、インターリーブオプションを使用したときの結果を表 2.4 に示す。インターリーブオプションを使用しない場合は、画質・音声ともに劣化するが、インターリーブオプションを使用することにより、画質・音声ともに改善した。しかし、誤り訂正コードを 3 個付与したとき画質・音声の劣化が見られ、4 個付与したときに急激に劣化した。

2.3.5 考察

実験結果として、次の 3 つの知見が得られた。

- FEC により、スループットは 30 ~ 45% に低下した。
- FEC は、パケットの損失がパルスのに発生する場合、非常に高い確率で復元できる。短いパーストがパルスのに発生する場合も、インターリーブオプションを用いることで復元率を高く維持することができた。
- 実トラフィックとして DV を転送した結果、FEC により画質・音声ともパケットの損失がない品質と同程度まで改善できた。しかし、過剰な訂正コードの付与は、逆に品質の劣化を生じた。

FEC はパケットのパルスの損失に対して有効である。さらに、短いパースト誤りに対してもパケットの送信パターンを Reorder するインターリーブオプションで復元率を維持できた。しかし、ネットワークが輻輳してきた時に発生する長いパースト誤りには、十分な誤り訂正ができない。訂正能力を高めつつ、ネットワークの輻輳を引き起こさないためには、付与する冗長パケット数、インターリーブする Block 数などのパラメータの動的な調整が必須である。実際のネットワークの利用可能帯域を動的に知り、それに応じたレートでパケットを送信できるような機構を考える必要がある。

DV 転送実験では、訂正コードを 4 個付与した場合、逆に、品質の劣化を生じている。これは、DV の $\frac{1}{2}$ レートに訂正コードを 4 個付与した場合のトラフィックは約 29Mbps であり、2.3.2 節のスループットの約 25Mbps を上回る。したがって、品質の劣化

の原因は FEC 処理が DV 転送処理に追いつかないためと考えられる。インターリーブオプションを使用した場合、パケットの Reorder のためにメモリに保持するパケット数が増加し、その結果、メモリ消費による FEC 復元処理の性能劣化を引き起こす結果となった²。また、FEC によりパケット数が大幅に増加することにより、ルータの動作負荷の上昇が懸念されたが、実験で用いた PC ルータは安定して動作し、FEC による DV の再生品質も高く維持できた。

2.4 Building Blocks 方式による Reliable Multicast

2.4.1 PGM 概要

PGM (Pragmatic General Multicast) は、Cisco Systems 社が 1998 年の IRTF で発表した Reliable Multicast プロトコルである [43]。PGM の主なパケット型の定義を表 2.5 に示し、損失データの信頼性保証の概略を図 2.10 に示す。

PGM は、損失パケットの再送を NACK (否定応答) により制御するプロトコルであり、パケットの到着順序の保証、重複の回避を可能とする。送信データの損失に対しては、受信ノードが管理責任を持つ。ACK を用いるプロトコルと比較して受信ノード数のスケーラビリティが確保できる。これは、NACK の発生頻度が ACK と比較して低いこと、ACK はデータの受信結果を送信ノードが管理するため、受信ノード数に応じた処理能力が送信ノードに要求さ

表 2.5. PGM パケット型の定義

パケット型	意味
SPM	Source Path Message. PGM ツリーを形成する
ODATA	Original Data. 配送するデータ
RDATA	Repair Data. 再送データ
NAK	Negative Acknowledgement. 否定応答
NCF	Negative acknowledgement ConFirmation. 否定応答受信確認

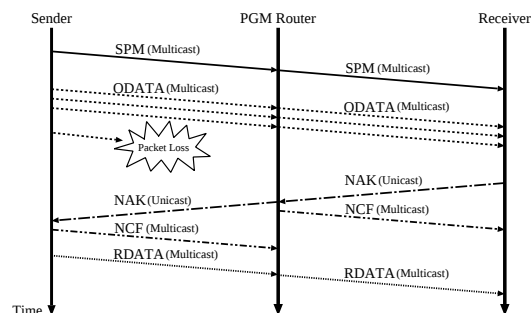


図 2.10. PGM データ転送

れるが、NACK はデータの受信結果の管理責任が各受信ノードに分散されていることが要因である。また、NACK が損失した場合に、送信ノードが正常受信したとみなされる問題に対しては、NCF (NACK に対する確認応答) を用いることにより NACK 自身の信頼性保証をしている。また、ルータの支援により NACK suppression を実現しているため、NACK implosion を回避し、受信ノード数のスケーラビリティが確保できる。PGM ルータは、下流の受信ノードからユニキャストで送信された NACK を受信した場合、NACK に対する NCF を下流にマルチキャストする。これにより、NCF を受信した他の受信ノードは NACK の送信を控える。また、PGM ルータは重複する NACK を受信した場合、上流に転送せずに集約を行う。

NACK-based プロトコルの問題点は、ACK-based プロトコルのような定期的なフィードバックがないため、インターネットへの適用時に最重要となる TCP Friendly な輻輳制御の実現が困難である点である。PGM は、これに対して ACK の役割を持つ NULL NACK パケットを送信するオプション機能を持つが、十分なスケーラビリティの実現には至っていない [141]。

2.4.2 Reliable Multicast の実装

PGM は、IPv4 対応の FreeBSD で動作するホストインプリメンテーションがオープンソースで配布されている [135]。本章では、これを IPv6 対応に修正し、エンドエンドでの Reliable Multicast の実現を可能としている。レイヤー構造を図 2.11 に示す。

² データの特性に合わせて選択的に FEC を適用することで、FEC 処理負荷の軽減、冗長帯域の増加の抑制が可能である [196]。

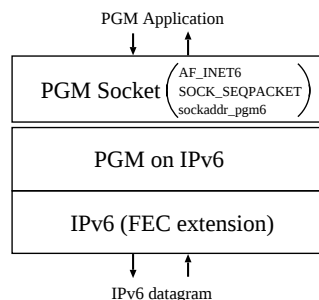


図 2.11. PGM-FEC on IPv6 レイヤー構造

2.4.3 評価項目

実装した FEC と PGM を、IPv6 マルチキャストプロトコルである PIM-SM に組み合わせて適用し、評価する。評価項目として、以下の 2 つを挙げる。

● 転送データの信頼性

PGM は、既に送信バッファから廃棄されたデータについては回復できない。回復に失敗した場合は、受信ファイルには欠損が生じる。転送データの信頼性の指標として、各々のパケット損失確率に対してファイルの欠損状態を調べること、転送データの受信成功確率を測定する。

● 再送・再送制御パケット数

ネットワーク輻輳の主な原因として再送・再送制御トラフィックの増加がある。FEC の Open Loop Delivery と、PGM の NACK-based Reliability は、再送パケットおよび再送制御パケットを減少させる。したがって、受信ノード数のスケーラビリティの向上が期待できる。しかし、FEC は誤り訂正コードにより冗長な帯域を増加させ、本章での IPv6 パケットの分割手法は、ネットワーク全体の総パケット数を大幅に増加

させる。現状のルータの動作は、帯域の増加よりパケット数の増加が高負荷であるため、その結果、ネットワーク輻輳を発生する可能性がある。再送制御パケットとして NAK と NCF、再送パケットとして RDATA の総数を測定する。

2.4.4 性能評価実験

実験環境を図 2.12 に示す。

送信ノード (Send) の計算機環境を表 2.6 に、受信ノード (Recv1, Recv2) の計算機環境を表 2.7 に示す。

性能評価は、PGM ソースコードとともに配布されている pgmcat コマンドを、IPv6 対応に修正して用いた。Send から Recv1 と Recv2 に 500KB のファイルを IPv6 マルチキャスト転送する。転送フレームサイズは 512byte と設定し、IPv6 マルチキャストプロトコルには PIM-SM を用いた。

各ルータで 1%～10%の確率でパケット損失させた場合の、転送ファイル欠損状態と、ネットワーク内に送信された ODATA、RDATA、NAK、NCF の総数を測定した。総数の測定は、PGM 用パッチをあてて IPv6 用に変更を加えた tcpdump コマンドを各ルータで使用する。

表 2.6. 計算機環境 (送信ノード)

OS	FreeBSD 3.4-RELEASE KAME 20000207-snap PGM kpgm-000216.tgz
CPU	500MHz (PentiumIII)
Memory	256MB
NIC	Ethernet 100Mbps (EtherExpress Pro/100+)

表 2.7. 計算機環境 (受信ノード)

OS	FreeBSD 3.4-RELEASE KAME 20000207-snap PGM kpgm-000216.tgz
CPU	600MHz (PentiumIII)
Memory	256MB
NIC	Ethernet 100Mbps (EtherExpress Pro/100+)

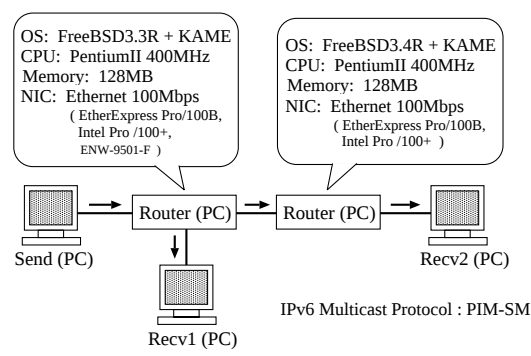


図 2.12. Reliable Multicast 実験環境

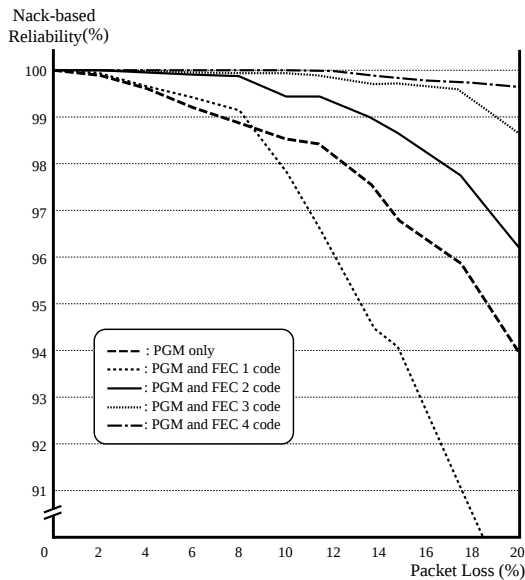


図 2.13. データ受信成功確率 (Recv2)

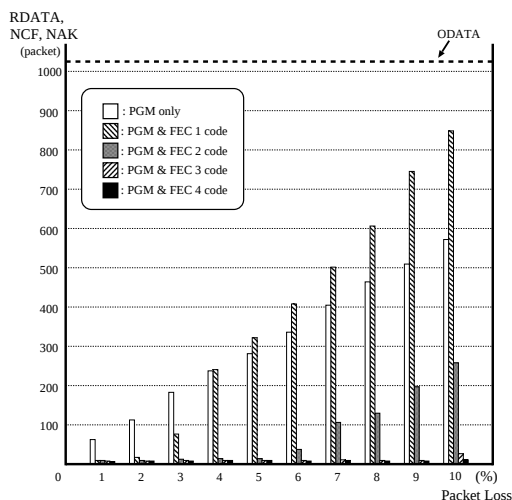


図 2.14. ODATA, RDATA, NAK, NCF 総数

実験結果

データの受信成功確率を図 2.13 に示す。RDATA、NAK、NCF の総数を図 2.14 に示す。

2.5 実験結果に対する考察

受信データの信頼性

図 2.13 より、PGM と FEC の組み合わせにより、転送データの受信成功確率が向上している。FEC の訂正コード数を 1 個付与した場合、パケット損失率が 3%以上、FEC の訂正コード数を 2 個付与した場合、パケット損失率が 8%以上で転送データの受信成功確率が低下する。これは、図 2.7 より、FEC の

復元率の低下に依存しており、PGM による性能劣化ではないことが分かる。

パケットの損失率が大きい場合、受信データは明らかに欠損している。NACK based のプロトコルは、送信ノードが転送データを一定時間保持しておくための負荷が高く、NACK を受信する前に、転送データを破棄した場合は復元に失敗する。そのため、パケット損失に対する信頼性は、マシン性能に大きく依存することが問題点として挙げられる。データ転送における完全な信頼性を提供するためには、FEC と PGM の組み合わせの他に、ADU (Application Data Unit) による何らかの信頼性保証機構等が必要となる。

RDATA、NAK、NCF 総数

500KB のファイルを 512byte フレームで転送した場合、ODATA は 1024 パケットとなる。図 2.14 より、PGM と FEC の組み合わせにより、ネットワークに送出された RDATA、NAK、NCF 総数の増加を抑制する結果が得られた。FEC の訂正コード数が十分であれば、RDATA、NAK、NCF 数をほぼ 0 に近い状態に押えることが可能である。したがって、PGM と FEC の組み合わせにより、受信ノード数に対するスケーラビリティを向上する Reliable Multicast の実現は可能である。

問題点

問題点としては、PGM がマシン性能や実験環境に大きく依存することが挙げられる。

- 本実験で、転送ファイルを 500KB と限定した理由は、PGM と FEC の組み合わせにより、多量のカーネル内メモリを必要とするため、500KB 以上のファイル転送はカーネル内で ENOBUF エラーが検出されたためである。
- PGM は、NACK 送出のインターバルや NCF と RDATA の受信のためのタイマー等、設定可能なパラメータを多く備えており、本実験では送信ノードと各受信ノードにおいて、ネットワークポロジやパケット損失率を考慮して特化した設定を行った。設定が的確でなければ、受信データの欠損率は大幅に上昇し、必要以上の性能劣化を生じる結果となった。

上記問題に対しては、再送要求パケット・再送データに重要度を合わせて選択的に FEC を使用する、pro-active 型 FEC から reactive 型 FEC へ移行する、などの FEC と PGM の各処理負荷の軽減により対応することが必要がある。

2.6 今後の課題

FEC と PGM による Reliable Multicast

本章では、Reliable Multicast への組み込みが標準化されている Protocol Component を組み合わせて実装しており、各機能をつなぐ Protocol Core や他の Building Block と組み合わせた実装と評価が課題として挙げられる。特に Protocol Core については、Internet Draft が 2000 年末に提出され、今後、実装による評価への要求が高まることが予想される。また、Reliable Multicast 標準化の必須条件とされている TCP Friendly 輻輳制御についてはまだ完全な解決方法が提案されておらず、完成度の高い Reliable Multicast の標準化のためには、実装による評価と考察が必要である。

また、FEC coding Building Block は、Layer4 でのパケットヘッダの統合を目的にトランスポート層で実装する方向で標準化が進んでいる。FEC のトランスポート層での実装は、ジッタの発生によりデータ品質が劣化する問題点を除けば、動的に訂正符号を選択、pro-active 型 FEC と reactive 型 FEC の選択、などの利点がある。しかし、レート調整による輻輳制御や、受信ノードからの通知方法など、Protocol Core へ大きく依存するため、標準化が待たれる。

各 Building Block の性能指標

現在、Building Blocks 方式の標準化は、アプリケーションやネットワークからの要求に着目して進められている。しかし、端末やネットワークに各 Building Blocks が要求する総合的な性能の評価には至っていない。

本章で実装した FEC と PGM による Reliable Multicast は、端末に高性能を要求したため、パケットの損失に対する信頼性を低下し、実験環境を限定する結果となった。Building Block の組み合わせに自由度を与え、様々な要求に順応した Reliable Multicast の実現するためには、個々の Building Block

が要求する性能の指標を示すことが重要である。

2.7 まとめ

本章では実際に Building Block として FEC と PGM を組み合わせて Reliable Multicast を実装し評価することで、各 Building Block および Reliable Multicast の問題点を明確化し、課題を抽出した。

組み合わせによる Reliable Multicast の問題点として、PGM はパケット損失に対する信頼性に関して、転送データを保持しておく送信ノードの性能に大きく依存することが挙げられた。さらに、FEC と PGM の組み合わせ実装は、端末に高性能を要求し、実験環境を限定する結果となった。今後の課題として、Reliable Multicast 標準化に合わせて実装していくこと、端末やネットワークに各 Building Block が要求する性能の指標を示し、総合的な性能を評価していく必要性を挙げた。

第 3 章 動的なプロトコルの選択を用いた高信頼性マルチキャスト通信の評価と応用

3.1 はじめに

近年の爆発的なインターネットの普及にともない、多くの新しいアプリケーションが開発されている。その中でも、遠隔会議や、ソフトウェアの電子的配布などは、1 対多、もしくは多対多の形で通信が行われるアプリケーションである。

このようなアプリケーションにおいて、1 対 1 の通信に用いられるユニキャストを適用した場合、送信者はパケットを受信者の人数分コピーし、各受信者に送信する必要がある。また、送信者は各受信者のアドレスを知っておく必要があり、送信者と、送信者付近のネットワークに大きな負荷がかかることになる。同様のアプリケーションに対し、マルチキャストを適用した場合には、送信側でデータを複製する必要がなく、送信されたデータは、ルータで経路が分岐する場合にのみコピーされるので、送信側の負担を軽減し、冗長なトラフィックの発生を抑えることができる。

このマルチキャスト通信の中で、IP (Internet Protocol) を通じて行われるのが IP マルチキャストである。IP マルチキャストでは、送信者が受信者のリス

トを維持する必要がなく、受信者はマルチキャストセッションに自由に参加、離脱することができるため、大規模で動的なグループでの通信が可能である。

ここで、マルチキャスト通信における信頼性について考えることにする。例えば大容量データの配信を行う場合、各受信者に送信された全てのデータが届くことは必須の条件である。このようなアプリケーションにおいては、「信頼性のある通信」が必要となる。ユニキャスト通信において信頼性が必要な場合は、トランスポート層プロトコルとして TCP (Transmission Control Protocol) を使用することができる。しかし、マルチキャスト通信では TCP を使用することができない。なぜなら、TCP では送信者が受信者からの ACK (肯定応答) を受け取る必要があり、マルチキャスト通信において複数の受信者から ACK を受信することを考えると、受信者が多数になった場合には過度のトラフィックが送信者に集中することになるからである。

そこで IP マルチキャストではトランスポート層プロトコルとして UDP (User Datagram Protocol) を使用している。最善努力型のデータ配信を行う UDP では、送信されたデータがグループに参加している全ての受信者に届くことや、送出された順序通りに届くことが保証されないため、現在はマルチキャストアプリケーションの中に、エラーの通知やデータの再送などの処理が実装されている。

現在、マルチキャスト通信に信頼性を提供する高信頼性マルチキャストプロトコル [10, 21, 108] は数多く提案されている。これらのプロトコルの中には、エラーの通知、パケットの再送の他にも、グループに参加している受信者の管理といった、マルチキャスト固有の問題に対処できるものもある。高信頼性マルチキャストプロトコルは、そのアプローチの仕方から次の 2 種類に分けることができる。

- 全てのマルチキャストアプリケーションに対して信頼性を提供できる汎用性のあるプロトコルを目指したもの
- ある特定のアプリケーションのみに対して信頼性を提供するプロトコルを目指したもの

前者は、1 つのプロトコルで、「マルチキャスト適応型の TCP」を目指すものである。しかし、マルチキャストアプリケーションからの要求は、規模適応

性、送出されたデータの順序の保存、遅延の少ない配送など様々であり、1 つのプロトコルで全てのアプリケーションの要求を満たすことは困難である。後者は、例えば少人数で行う対話型アプリケーションに対しては、規模適応性を犠牲にして遅延の少ないプロトコルを採用するというように、あるアプリケーションに特化したプロトコルを作成するというものである。しかし、このようなプロトコルのほとんどがアプリケーションで実装されており、新たなアプリケーションに対して既存のプロトコルを適用する場合に、移植が難しいというような応用の面で問題があると言える。

3.2 本研究の目的

3.2.1 汎用性のある高信頼性マルチキャスト通信の実現

3.1 節において、現在のマルチキャスト通信が抱える問題点について述べた。汎用性のあるプロトコルを目指すというアプローチの場合は実現が難しく、アプリケーションに合わせたプロトコルについては応用の面で問題がある。そこで我々の研究室では、汎用性のある高信頼性マルチキャスト通信を実現するために、マルチキャスト通信に信頼性を提供する処理をアプリケーションに依存しない形で行うための手法を提案している。具体的には次の 3 つの処理が挙げられる。

- (1) 今までアプリケーションで高信頼性マルチキャストプロトコルが行っていた処理をネットワークに組み込む。
- (2) アプリケーション側で使用するプロトコルを選択する必要はなく、ネットワークが用途、環境に合わせて適切なプロトコルを選択する。
- (3) 動的に変化するネットワークに対応して、複数の高信頼性マルチキャストプロトコルを組み合わせ使用して使用する。

(1) は現在アプリケーションで実装されている高信頼性マルチキャストプロトコルをアプリケーションから切り離し、ネットワークの各ノードにプロトコルスタックを組み込んで、信頼性を提供する処理を行うというものである。これにより、アプリケーションと高信頼性マルチキャストプロトコルとの間に依存関係がなくなり、新しいアプリケーションへ

の対応も容易になる。

(1) によってアプリケーションと高信頼性マルチキャストプロトコルが切り離されると、各アプリケーションにおいてプロトコルを選択することが必要になる。そこで、(2) によってアプリケーションの種類やマルチキャストグループの規模、ネットワークの状況などからネットワークが自動的に適切なプロトコルを選択して、信頼性を提供できるようにする。(1)、(2) によって、アプリケーションは信頼性やグループに参加している受信者の管理などを考慮せずに、送信、受信などの簡単な処理を行うだけでよいことになる。

(3) の「ネットワークの変化」とは、時間的、空間的なネットワークの変化を指している。時間的な変化への対応とは、例えば、グループでのセッション中に受信者の数が増えてきた場合には、規模適応性のあるプロトコルに切り替えるといった形で行われる。また、空間的な変化への対応とは、送信者に比較的近い受信者はエラーを発見した際、送信者と直接メッセージを交換して再送処理を行い、送信者と離れている受信者は近くの受信者同士でメッセージを交換することにより、局所的にエラーを修復するというように、ネットワーク上の送受信者の位置や輻輳が発生している場所によって、選択するプロトコルを変えていくというものである。

3.2.2 本研究の目的

我々の研究室で提案している新しい手法を実装する際には、状況に応じた適切なプロトコルの選択が重要となる。そこで本研究では、まず最初に高信頼性マルチキャストプロトコルの評価を行い、プロトコルの長所、短所を明らかにすることによって、プロトコルの選択を行う際の大まかな指針を得る。

次に、プロトコルの評価を行った結果を用いて、各プロトコルを単独で使用してマルチキャスト通信を行った場合と、状況に応じて複数のプロトコルを組み合わせで通信を行った場合との通信効率の比較を行い、今後実装する予定である本方式の効果について述べる。

3.3 高信頼性マルチキャスト通信の評価

3.3.1 高信頼性マルチキャストプロトコル

本研究では、数多く存在する高信頼性マルチキャストプロトコルの中で、AFDP[28]、RMTP[126]、

SRM[49] という 3 種類のプロトコルに注目した。この 3 種類のプロトコルの特徴を簡単に述べると、再送制御については、AFDP は送信者からのマルチキャストによる再送、RMTP は DR(Designated Receiver) による局所的な再送、SRM は送信者、受信者の両者がマルチキャストによる再送を行う。輻輳制御については、AFDP は受信者がデータの欠落を発見した場合に送信帯域を下げ、RMTP は TCP のスロースタート [147] と同じような手法を用いている。SRM については、データの送信帯域を下げるといった処理は行われていない。

本研究でこの 3 種類のプロトコルに注目したのは、これらのプロトコルは特徴が明確であり、対象とするアプリケーションや適用できるグループの規模などが異なるため、あるプロトコルの短所を他のプロトコルで補うことが容易になり、複数のプロトコルを組み合わせることによる効果が現れやすいと考えたからである。

3.3.2 評価の方法と評価基準

高信頼性マルチキャストプロトコルを評価する場合、考慮すべき項目は非常に多い。そこで本研究では、評価を行う際にシミュレーション [180] を用いた。シミュレーションを用いることには、次のような利点がある。

- 各リンクに対してパケット損失率を設定することが可能であり、例えば、「送信者に近い場所でエラーが多く発生している場合」などのような細かいネットワークの状況を設定することが可能である。
- 受信者の数や、パケット損失率、送信されるデータの種類などのパラメータを容易に変更することができ、様々なパターンでの実験を行うことが可能である。
- パケット数やパケットの到着時刻などの値を一元的に管理することができ、これらの値を用いてプロトコルの性能を定量的に評価することが可能である。

本研究では、シミュレーションを用いることによる以上のような利点をふまえて、種々のパラメータの値を変更しながら数多くの実験を行い、各高信頼性マルチキャストプロトコルの長所、短所を明らか

にする。

本研究では、プロトコルの評価を行うにあたり、各プロトコルの再送制御と輻輳制御の効率の良さに注目した。以下に、再送制御、輻輳制御の説明と、評価の際にこれらに注目した理由を示す。

再送制御

ネットワーク上に何らかの障害が発生し、受信者が全てのデータを受け取ることができなかった場合、送信者はもう一度欠落したデータを送り直す必要がある。これが、データの再送である。3.3.1 節で述べた 3 種類のプロトコルは、全てマルチキャストを用いたデータの再送を行うため、正しくデータを受け取った受信者に対しても再送されたデータが届く可能性がある。このような重複したデータが数多く送信されると、ネットワーク上に冗長なパケットが数多く流れることになり、輻輳の原因となる。そこで本研究では、各受信者に届いた再送パケットの数をカウントし、このパケット数が少ない程、効率が良く、性能の良いプロトコルであるとして、比較、評価を行う。

輻輳制御

ネットワーク上のある場所において輻輳が発生した場合、もしくはデータの配信中にパケットの欠落が発見された場合には、データの再送を行う可能性が高く、ネットワーク上に流れるパケット数が多くなる。このデータの再送によって、新たな輻輳の発生を防ぐために、高信頼性マルチキャストプロトコルの中には輻輳制御を行うものが多い。これは、エラーを発見した際に、データの送信間隔を大きくするなどして、ネットワーク上に流れるパケット数を少なくするための処理である。輻輳制御が効率良く行われると、ネットワークの状態が比較的良好な時には、データを速く配信することができ、状態が悪い時には、データの送信帯域を下げることにより、輻輳を避け、結果的にデータを速く配信することができる。本研究では、一定のサイズのファイルを配信し終わるのにかかる時間を測定し、スループットを算出する。このスループットが高い程、効率の良い輻輳制御が可能で、性能の良いプロトコルであるとして、比較、評価を行う。

3.4 シミュレーションによる高信頼性マルチキャストプロトコルの評価

3.4.1 ns2

本研究では、シミュレーションを行うためのソフトとして、UCB(カリフォルニア大学バークレー校)で作成されたネットワークシミュレータ ns2[117]を使用した。ns2 は、マルチキャスト通信のシミュレーションにも対応しており、シミュレータに新たなプロトコルを組み込むことができるため、今回注目した高信頼性マルチキャストプロトコルについても、AFDP と RMTP についてはある程度のモデル化を行った後シミュレータに組み込むという作業を行った。SRM については ns2 に組み込まれていたものを利用した。

3.4.2 想定したネットワーク

今回のシミュレーションでは、「日本全国に存在する受信者に対して、データの配信を行う」程度の規模を想定した実験を行う。そこで、より現実に近い環境での実験を行うために、JGN (Japan Gigabit Network)[84] のギガビットネットワーク通信回線を参考にしながらネットワークのトポロジや帯域を設定した。JGN とは、通信・放送機構が、超高速光ファイバ通信網 (ギガビットネットワーク通信回線) と共同利用型研究開発施設 (ギガビットラボ) を整備し、大学、研究機関等に開放するものであり、ギガビットネットワーク通信回線は全国 10 ケ所の ATM 交換設備及び 39 ケ所の接続装置を超高速光ファイバで結んだ最大 2.4Gbps の帯域をもつ通信回線である。

本実験で扱うネットワークは、JGN のバックボーンの部分を主に参考にし、バックボーンルータにつながっている末端の部分については、簡単のために各ルータに同数のノードをつなげて、トポロジは単純な木構造とした。このネットワークを、図 3.1 に示す。

図 3.1 において、合計 143 個あるノードは受信者となることができ、図中の中央部にある四角で囲まれたノードを送信者としている。本実験では、末端以外のノードについては、受信者とルータの役割を兼ねることができる。リンクについては、現在 JGN がもつ通信帯域を参考にし、さらに近い将来に実現されると考えられる超高速インターネットを考慮して、最も太い線がバックボーンとなる 10Gbps の全

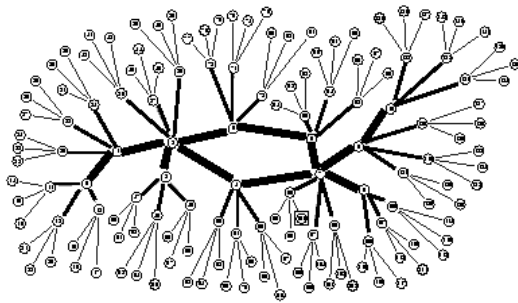


図 3.1. 実験環境

二重線、次に太い線が 1Gbps の全二重線、最も細いが 100Mbps の全二重線となっている。

3.5 再送制御の評価

3.5.1 実験 1：送信されるデータの種類による比較

まず実験 1 では、送信されるデータの送信帯域を変えた場合の各プロトコルの比較を行う。この実験により、各マルチキャストアプリケーションに対してどのプロトコルを適用すれば良いかを知ることができる。実験では、図 3.1 で示したネットワークにおいて、100 の受信者をグループに参加させ、データの配信を行う。また、受信者付近にエラー発生リンクを設置し、パケットの損失率は、リンクを通るパケットの総バイト数の 5% としている。

このようなネットワークにおいて、送信者から次の 2 種類のデータを送信した場合に、再送パケット数にどのような変化がみられるかについての比較を行った。送信するパケット数は 1000 個としている。

- 送信帯域 10Mbps、パケットサイズ 1000 バイト
- 送信帯域 30Mbps、パケットサイズ 1000 バイト

実験結果を表 3.1 に示す。表中の数値は、各プロトコルを使用した場合に各受信者が受け取る再送パケット数の 1 受信者あたりの平均である。まず、10Mbps、1000 バイトのデータを送信した場合、表からは、RMTP が最も良い性能を示していると言える。次に AFDP、SRM の順となっている。このように 1 受信者が受け取る再送パケット数の平均を考

表 3.1. データの送信帯域による再送パケット数の比較

	AFDP	RMTP	SRM
送信帯域：10Mbps	476	19	587
送信帯域：30Mbps	4014	18	594

えた場合、局所的な修復を行う RMTP では再送パケットを全く受信しない受信者も存在するため、パケット数が非常に少なくなっている。

30Mbps、1000 バイトのデータを送信した場合、RMTP については 10Mbps、1000 バイトのデータを送信した場合と同様に、最も良い性能を示している。しかし、AFDP と SRM については、性能の順が入れ替わっており、SRM については 10Mbps、1000 バイトの時と比較しても性能に変化がないのに対し、AFDP については極端に性能が悪くなっている。AFDP は、再送の必要が生じた場合、直接送信者との間で再送制御を行うため、送信者付近のリンクにトラフィックが集中しやすい。それに加えて、今回は高速なデータ配信を行っていたため、送信者付近のリンクに輻輳が生じる可能性が非常に高くなっていた。実際、実験の経過をアニメーションで見ていると、送信者付近のリンクにおいて輻輳が観測され、パケット損失率が極端に大きくなっていた。輻輳が生じると、データを正しく送れないため、さらに再送の必要が生じ、リンクを圧迫するという悪循環に陥っていたと考えられる。AFDP の輻輳制御については、効果がみられなかった。

3.5.2 実験 2：エラーが発生するリンクの場所による比較

RMTP のような局所的な修復を行うプロトコルでは、エラーの発生箇所によってプロトコルの性能が変化すると予想される。そこで実験 2 では、パケットロスを発生するリンクの場所を変えた場合の各プロトコルの比較を行う。エラー発生リンクの場所については、受信者付近に存在する場合と、送信者付近に存在する場合の 2 通りを考えた。

送信者付近でエラーが発生する場合については、送信者から送出されたパケットが、バックボーンに到達するまでに集中してパケットをロスするような設定になっている。送信者付近でパケットロスが多く発生すると、データを正しく受け取れない受信者が多くなるため、AFDP では送信者へ NACK が集中し、性能が悪くなることが予想される。また、RMTP では、DR までデータが到達しないことが多く、局所的な修復が機能しないため、再送が送信者との間で行われることになり、性能が低下すると予想される。

実験では、受信者は実験 1 と同じく 100 に設定し、送信するデータは送信帯域 10Mbps、パケットサイ

表 3.2. エラーが発生する場所による再送パケット数の比較

	AFDP	RMTP	SRM
受信者付近でのエラー	476	19	587
送信者付近でのエラー	1368	21	111

ズ 1000 バイトとし、パケット数は 1000 個とした。エラー発生リンクのパケット損失率は 5% である。

表 3.2 に実験結果を示す。受信者付近でエラーが発生する場合については、実験 1 の 10Mbps の場合と同様である。送信者付近でエラーが発生する場合、AFDP については、予想通り、データを受け取れない受信者が多く存在するため、再送パケット数が多くなっている。RMTP については、パケットの数についてはそれほど多くなっていない。これは、RMTP の ACK の集約により、同じデータに対する再送が重複して行われるのを防いでいるからである。SRM については、受信者付近でロスが多い場合よりも、送信者付近でロスが多い場合の方が再送パケット数が少なくなっている。

3.5.3 実験 3：規模適応性の比較

一般的に、AFDP は規模適応性のないプロトコルと言われており、RMTP、SRM は規模適応性があると言われていた。そこで実験 3 として各プロトコルの規模適応性の比較を行うために、図 3.1 のネットワークの末端に、木構造のネットワークを足していき、500 のノードを持つネットワークを新たに作成した。このネットワークにおいて、受信者を 100 ～ 500 まで変化させた場合に、再送パケット数がどのように変化するかを測定する実験を行う。

実験では、受信者付近に 5 ケ所のエラー発生リンクを設定し、パケット損失率は 5% とした。送信するデータは送信帯域 10Mbps、パケットサイズ 1000 バイトとし、パケット数は 1000 個とした。

実験結果を図 3.2 として示す。このグラフの横軸はグループに参加している受信者の数、縦軸は受信者が受け取った再送パケット数の平均である。局所的な修復を行う RMTP については、1 受信者あたりの平均をとっているためパケット数は非常に少なくなる。RMTP の性能はグループの規模が大きくなった場合にも変化していない。これに対し、AFDP では規模が大きくなるにつれて再送パケット数も多くなり、受信者が 100 ～ 110 の間は SRM よりも良い

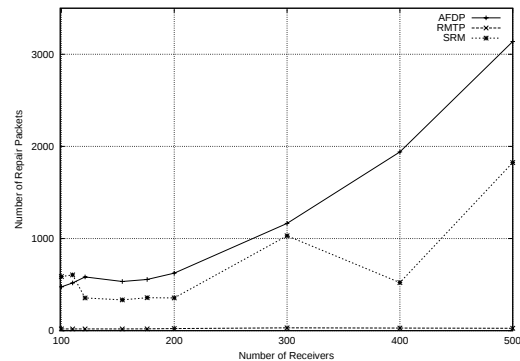


図 3.2. 規模適応性の比較

性能を示しているのに対し、規模が大きくなると性能が逆転している。SRM については、受信者が 300 のところで再送パケット数が多くなっている。これは、SRM の性能がグループの規模だけでなく、受信者の分布の仕方にも密接に関わっていることを示している。

3.5.4 実験 4：輻輳制御の評価

実験 4 では、10 メガバイトのファイルを配信し終えるのにかかる時間を測定し、スループットを算出することによって、各プロトコルの性能を比較する。エラーが発生するリンクの場所について、受信者付近でパケットをロスする場合と送信者付近でロスする場合の 2 種類のパターンを考え、各リンクのパケット損失率を 0 ～ 10% まで変化させる。送信するデータは、送信帯域 10Mbps、パケットサイズ 1000 バイトとし、10 メガバイトを配信し終えた時刻を計測して、1 秒あたりのスループットを算出した。

分散型マルチメディアアプリケーションに対応している SRM については、データを配信する際の遅延も少ないと予想される。AFDP については、送信者付近でエラーが発生した場合に再送パケット数が非常に多くなるという結果が実験 2 から得られており、受信者がデータの欠落を発見した場合に送信帯域を下げる仕組みでは、スループットが極端に低下するのではないかと予想される。

実験結果を図 3.3 として示す。このグラフの横軸は各リンクのパケット損失率 [%] であり、縦軸はスループット [Mb/sec] である。まず、受信者付近にエラーが発生する場合、RMTP と SRM については、パケット損失率が高くなるにつれて、スループットがかなり低下している。これに対し、AFDP は他の 2 種類のプロトコルと比べて、性能の低下がそれほ

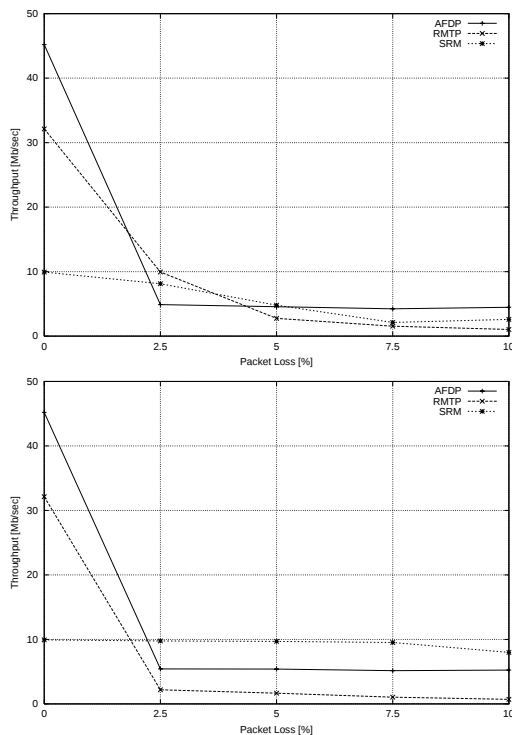


図 3.3. 輻輳制御の効率の比較

ど大きくない。送信者付近にエラーが発生する場合については、実験 2 でも良い性能を示した SRM が、パケット損失率が高くなっても性能の低下を生じないという結果になっている。次に性能が良かったのが AFDP であり、RMTP は他の 2 種類のプロトコルと比べて性能の低下が非常に大きかった。

本実験では、エラーが発生する場所に関係なく、RMTP が性能が悪く、AFDP は予想に反して良い性能を示した。これは輻輳制御の方法の違いが原因である。RMTP は送信者に返ってきた ACK、もしくは再送要求によってデータの送信帯域を決定する。再送要求がある閾値を超えた場合には、ウィンドウ

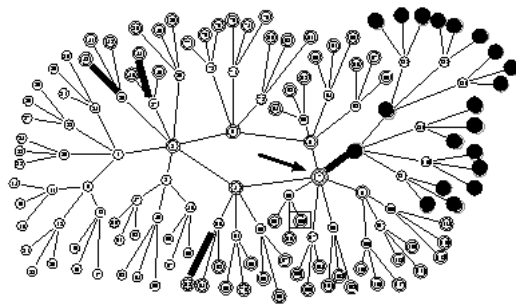


図 3.4. プロトコルの分布状況 (● : SRM、○ : RMTP)

サイズを 1 に設定し、輻輳を回避しようとする。このスロースタートの手法により、データの送信に時間がかかってしまう。これに対し AFDP では、受信者から NACK が返って来なければデータの送信時に送信帯域を上げていく。この手法では、受信者からの NACK によって一時的に送信帯域が下がった場合でも、データの送信を行うたびに送信帯域が上がるため、回復が早い。このような理由から AFDP の性能が高いという結果が得られたと考えられる。

3.6 実験結果の高信頼性マルチキャスト通信への応用

3.6.1 複数のプロトコルを組み合わせることによる効果に関する考察

3.2.1 節で述べたように、汎用性のある高信頼性マルチキャスト通信の実現には、複数のプロトコルを組み合わせることが必要になる。そこで複数のプロトコルを使用した通信の例として、ネットワークの空間的な変化を考え、場所によってプロトコルを使い分けることにより通信効率上がることをシミュレーションを用いて示す。使用したネットワークは、3.4 節の実験で用いたものと同様であるが、3.4 節での実験と異なるのは、エラーが発生するリンクの場所を太い線で示している。エラーリンクは末端に近いところに 3 ヶ所と、バックボーンのリンクに 1 ヶ所設定しており (パケット損失率は 5%としている)、受信者に近いリンクと送信者に近いリンクの両方にエラーが発生することになる。この場合、送信者 (図 3.4 中四角で囲まれたノード) から見てバックボーンに設定したエラー発生リンクより先では、データを正しく受け取れない受信者が多くなり、その他の場所についてはほとんどの受信者が正しくデータを受信できているという状態になる。そこで、3.4 節の実験で得られた結果から、ネットワークの状態によって複数のプロトコルを分布させることを考える。この分布の状況を図 3.4 として示す。

図 3.4 に示しているように、エラーが発生するバックボーンよりも先の受信者 (図 3.4 中では黒で塗りつぶしたノード) に対しては、送信者付近のエラーに対してスループットの低下がみられなかった SRM を使用し、他の受信者 (図 3.4 中二重丸で囲まれたノード) については局所的な修復を行う RMTP を使うことによって冗長な再送パケットの発生を抑える。ここで、送信者から見てバックボーンのエラー

表 3.3. 複数のプロトコルを組み合わせることによる効果

	RMTP	SRM	組合せ
再送パケット数の平均	253	4859	176
スループット [Mb/sec]	2.49	5.27	4.41

リンクの直前のノード (図 3.4 中では矢印で指されたノード) は、RMTP → SRM への変換を行うゲートウェイとなり、送信者から送られてくるデータを RMTP の受信者として受け取り、受け取ったデータを SRM の送信者としてエラーリンクより先の受信者に送信する。

複数のプロトコルを用いた場合と、RMTP、SRM を単独で用いた場合について、送信帯域 10Mbps、パケットサイズ 1000 バイトのデータを 10 メガバイト (約 10,000 パケット) 送信し、各受信者が受け取った再送パケット数と、スループットの測定を行った。

実験結果を表 3.3 に示す。表を見ると、再送パケット数については、RMTP、SRM を組み合わせた場合が最も良い性能を示しており、スループットについても、RMTP 単独の場合よりも良い結果が得られている。この実験結果から、RMTP、SRM を組み合わせて使用することにより、RMTP の局所的な修復による効率の良い再送と、SRM の高いスループットという 2 種類のプロトコルの長所を生かした効率の良い通信が行えるということが分かった。

本実験のように、アプリケーションの種類や、ネットワークの状況に応じて、複数の高信頼性マルチキャストプロトコルを組み合わせることによる効果は非常に高いと言える。

3.7 おわりに

本研究では、汎用性のある高信頼性マルチキャスト通信の実現を目指し、高信頼性マルチキャストプロトコルの選択を効率良く行うために、データの種類や、ネットワークの状況などの様々な要素を考慮しながら各プロトコルの評価を行った。この評価によって、各プロトコルの長所、短所が明らかになり、状況に応じて最適なプロトコルの選択を行うための指標を得ることができた。

また、3.5.1 節で行った、複数のプロトコルを組み合わせ使用した場合の通信効率を測定する実験では、ネットワーク上のエラーの発生箇所によって複数の高信頼性マルチキャストプロトコルを分布させ

ることにより、それぞれのプロトコルを単独で利用した場合に比べて非常に効率の良い通信を行えることが示された。複数のプロトコルを組み合わせ使用することは、各プロトコルの長所を生かすことのできる効果的な手法であると言える。

今後は、我々の研究室で提案している汎用性のある高信頼性マルチキャスト通信を実現し、3.5.1 節の実験で得られたような効率の良い通信が実際に行えるのかどうかについて検討する予定である。しかし、実装にあたって解決すべき問題点は多い。3.5.1 節の実験では、プロトコルの変換を行うゲートウェイでの変換処理に要する時間は考慮していない。実際にこの手法を用いた場合には、プロトコル変換のオーバーヘッドにより、スループットが低下することが考えられる。この手法を実装する際には、プロトコルの変換処理に要する時間を最小限に抑え、性能の低下を防ぐ必要がある。

本研究ではネットワークの状態により、場所によって複数のプロトコルを切り替えるという、ネットワークの空間的な変化に注目した実験、考察を行った。これに対し、時間的なネットワークの変化によって複数のプロトコルを切り替えて通信を行う場合には、次のような問題点を考慮する必要がある。

- 各プロトコルは通信を始める前に送信者へデータ送信の許可を与えたり、各受信者を管轄する DR を決定するといった準備が必要であり、途中でプロトコルを切り替える場合にはこれらのオーバーヘッドにより、スループットが下がる可能性がある。
- プロトコルが切り替わる時点において、エラー修復などの処理が途中だった場合、これらの処理を異なるプロトコルの間で引き継ぐ必要がある。

今後は以上のような問題点を考慮しながら提案している手法の実装を行い、実際のネットワークを使った実験を行う。具体的には、ファイル転送を行うアプリケーションを想定し、1 人の送信者から、多地点に渡る受信者に対してマルチキャストによるデータ配信を行う。この際に、ネットワークの状態を監視しておき、状況に応じて複数のプロトコルを組み合わせた通信と、各プロトコルを単独で利用した通信との効率の比較を行うことにより、先に述べたオーバーヘッドの影響などについて検討していく予定であ

る。それと同時に、様々な要素を考慮に入れた環境を作り出すことができるシミュレーションを今後も活用し、実装の参考にしていきたいと考えている。