

# プロトコル選択方式による高信頼性マルチキャスト通信の評価

柴田 賢介<sup>†</sup> 岡村 耕 二<sup>††</sup> 荒木 啓二郎<sup>†††</sup>

インターネット上において1対多, 多対多型の通信を効率良く行うための技術として, IP マルチキャストが注目され, 研究が進められている. IP マルチキャストでのデータ配信は, トランスポート層プロトコルとして UDP (User Datagram Protocol) を使用しており, 通信を行う際に信頼性が求められる場合には, 高信頼性マルチキャストプロトコルが各アプリケーションで実装されている. 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標とし, 再送制御やフロー制御など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. この手法の実現には, 用途, 環境に応じた適切な高信頼性マルチキャストプロトコルの選択を行う必要がある. そこで本稿ではプロトコル選択の際の指針を示すために, シミュレーションを用いて高信頼性マルチキャストプロトコルを評価する実験を行った. また, この実験結果の応用として, 高信頼性マルチキャストプロトコルを用途や環境に応じて組み合わせて使用した場合と各プロトコルを単独で使用した場合との比較を行い, 複数のプロトコルを組み合わせることによって通信効率が向上するという結果を得た.

## Evaluation of Reliable Multicast Transport with Protocol Selection Architecture

KENSUKE SHIBATA,<sup>†</sup> KOJI OKAMURA<sup>††</sup> and KEIJIRO ARAKI<sup>†††</sup>

IP multicast is an effective technology on the one-to-many and many-to-many communications. Multicast applications use UDP (User Datagram Protocol) for transport protocol and have the responsibility for reliable communications. Then it is difficult to apply these implementations to other applications, because implementations of multicast protocols have been pushed into application programs. We have proposed the reliable multicast architecture which pulls out the process for reliability from multicast applications. We have designed the mechanism of protocol selection which is the main part of our architecture. It is important for this mechanism to select the appropriate protocol to the condition of the network. In this paper, We have indicated the guide of protocol selection by evaluating performances of reliable multicast protocols using simulator, and estimated the effect of protocol selection.

### 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. その中には, 遠隔会議や, ソフトウェアの電子的配布などのグループ活動を主眼としたアプリケーションも含まれるが, このようなアプリケーションに対してユニキャストを適用すると, 送信者はパケットを受信者の

人数分コピーし, 各受信者に送信しなければならない場合がある. 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, 送信者の負担は大きい. 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, 送信者はデータを複製する必要がなく, 送信する際にはグループごとに割り当てられるマルチキャストアドレスにデータを送信すればよい. また, 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, 送信者の負担を軽減し, 冗長なトラフィックの発生を抑えることができる.

マルチキャスト通信の中で, IP (Internet Protocol) 上でデータをやりとりするのが IP マルチキャスト<sup>1)</sup> である. IP マルチキャストには, 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になるという

<sup>†</sup> 九州大学大学院システム情報科学府

Graduate School of Information Science and Communication Engineering, Kyushu University

<sup>††</sup> 九州大学情報基盤センター

Computing and Communications Center, Kyushu University

<sup>†††</sup> 九州大学大学院システム情報科学研究院

Graduate School of Information Science and Communication Engineering, Kyushu University

特徴がある。しかし、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 ) を使用することが多く、送信されたデータがグループに参加しているすべての受信者に届くことや、送出された順序どおりに届くことが保証されない。そこで現在は、マルチキャストアプリケーションの中に再送制御やフロー制御が実装されている。再送制御は受信しているデータにエラーが発生した場合の通知やそのデータの再送を行い、フロー制御ではデータ配信のスループットを上げるためにネットワークの状態に応じて送信者が送信帯域を調節する処理が行われる。

本研究では、このような処理をアプリケーションから切り離し、汎用性のある高信頼性マルチキャスト通信を実現することを目標としている。本稿では実現に際して必要となる高信頼性マルチキャストプロトコルの評価と、複数のプロトコルを使用することによる効果に関する考察を行った。まず 2 章において、本研究の目的と関連研究について述べ、3 章において各高信頼性マルチキャストプロトコルの評価の方法、基準などについて述べる。4 章ではシミュレーションを用いたプロトコルの評価の結果を示し、5 章では 4 章の実験で得られた結果の応用として、複数のプロトコルを使用した場合の効果について述べ、6 章をまとめとする。

## 2. 本研究の目的

### 2.1 本研究の目的

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は数多く提案されている<sup>1)</sup>。これらのプロトコルの中には、エラーの通知、パケットの再送のほかにも、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。たとえば、ファイル転送向けの AFDP<sup>2)</sup> というプロトコルでは、送信者、受信者のほかに、受信者の中から「秘書」と呼ばれる代表者を選ぶ。この秘書は送信者がデータを送る際の許可を与えたり、新たな受信者がグループに参加した場合の管理を担当したりする。

しかし、現在の高信頼性マルチキャスト通信には問題点がある。高信頼性マルチキャストプロトコルのほとんどはアプリケーションと一体化されて実装されており、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、再送制御やフロー制御のみをアプリケーションから取り出すということが難しく、移植性に欠けるという点である。そこで我々は、現在の高信頼性マルチキャスト通信がかかえるこのよう

問題点を解決するために、汎用性のある高信頼性マルチキャスト通信の実現<sup>3),4)</sup>を目標として研究を進めている。実現のための方針は以下の 2 つである。

- (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。また、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で使用する。

(1) は高信頼性マルチキャストプロトコルの処理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、再送制御やフロー制御を行うというものである。(1) により、プロトコルの処理がネットワークに組み込まれると、各アプリケーションはプロトコルを選択しなくなるとなる。そこで (2) によってマルチキャストグループの規模や、アプリケーションの種類に応じて、ネットワークが自動的にプロトコルを選択する。また、ネットワーク上の場所によるエラー発生の方の違ひに対応して、複数のプロトコルを分布させる。

現在、IETF の RMT ワーキンググループにおいて Building Blocks 方式<sup>5)</sup>が提案されており、高信頼性マルチキャストプロトコルの標準化作業が進められている。この Building Blocks 方式では、各高信頼性マルチキャストプロトコルが持つ機能に注目し、これらの機能を組み合わせることによって、各アプリケーションに合ったプロトコルを提供するものである。このように、高信頼性マルチキャスト通信では、現在既存のプロトコルの長所を生かして効率の良い通信を提供するという考え方がある。本研究において通信を開始する前に状況に応じた適切なプロトコルを選択という手法は、Building Blocks 方式の考え方に沿ったものである。

しかし、この方法では広範囲に広がるグループにおいてデータの配信を行う際に、ネットワークの状態が場所によって大きく異なった場合、通信効率が下がってしまうことが考えられる。そこで、ネットワーク上の場所に応じて複数のプロトコルを割り当て、効率の良い再送制御を行うことで、ネットワーク上に冗長なトラフィックが流れることを防ぐ。また、効率の良いフロー制御を提供することにより、全受信者に対して高いスループットを保証し、特定の受信者だけがデータの受信に時間がかかるという状況を改善する。

この手法を実装する際には、マルチキャストグループの規模やネットワーク上でのエラーの発生状況に応じて適切にプロトコルを選択する必要がある。そこで本稿では、先に述べた2つの方針のうち、(2)に注目し、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指針を得る。さらに、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る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。

## 2.2 関連研究

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のために、高信頼性マルチキャストプロトコルの処理をアプリケーションから切り離し、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を考えている。汎用性のある高信頼性マルチキャスト通信の実現に関する研究はこれまでいくつか行われており、例として以下のような研究があげられる。

2.1 節で述べたように、文献5)において Building Blocks 方式が提案されており、各高信頼性マルチキャストプロトコルが持つ機能を組み合わせることによって、各アプリケーションに合ったプロトコルを提供する手法が考えられている。この方式に対し、文献6)では Building Blocks 方式の機能の中で重要な要素となる FEC (Forward Error Correction) を実装し、既存の高信頼性マルチキャストプロトコルと組み合わせるための足掛かりとしている。

また別のアプローチとしては、アクティブネットワークの概念を利用した研究も進められている。文献7)で提案されている RMAC (Reliable Multicast using Adaptive Caching) という方式では、ルータに NACK や再送パケットの集約の機能と、キャッシュの機能を持たせるものである。マルチキャスト通信では中間ノードでキャッシュを行う場合、グループの数が多くなるとキャッシュの容量をオーバーすることが考えられるが、RMAC では輻輳の状況を監視しながら複数のルータで分散してキャッシュを行うことにより、容量オーバーの問題を回避している。

これらの研究と本研究が大きく異なる点は、空間的なネットワークの変化に対応できるという点である。単一のプロトコルで信頼性を提供する場合とは異なり、ネットワークの輻輳箇所などから、ネットワーク上の各地点に最適な複数の高信頼性マルチキャストプ

ロトコルを割り当てていくことにより、各プロトコルの長所を生かした効率の良い通信を実現できると考えている。

## 3. 高信頼性マルチキャスト通信の評価

### 3.1 高信頼性マルチキャストプロトコル

本稿では、AFDP<sup>2)</sup>、RMTP<sup>8)</sup>、SRM<sup>9)</sup>という3種類の既存の高信頼性マルチキャストプロトコルに注目した。この3種類のプロトコルの特徴について、再送制御とフロー制御をまとめたものを以下に示す。

- AFDP (Adaptive File Distribution Protocol)
  - 送信者からのマルチキャストによる再送
  - 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てフロー制御
- RMTP (Reliable Multicast Transport Protocol)
  - DR (Designated Receiver) による局所的な再送 (ACK Aggregation)
  - TCP (Transmission Control Protocol) のスロースタート<sup>10)</sup>に似たフロー制御
- SRM (Scalable Reliable Multicast)
  - 送信者、受信者の両者がマルチキャストによる再送
  - フロー制御は特になし

AFDP は NACK ベースのプロトコルであり、受信者の管理や再送は受信者の代表や送信者が行う集中管理型である。これに対し、SRM は同じ NACK ベースであるが、再送制御は受信者間でも行われ、誰か1人が管理するという形ではない。RMTP は ACK ベースのプロトコルである。ACK はデータを正しく受信した場合にも送信されるため、RMTP では DR と呼ばれる代表受信者を準備し、ACK パケットの集約を行うことによって、規模適応性を実現している。このほかに、高信頼性マルチキャストにおいてよく使われる手法としては FEC が考えられるが、本稿では IP パケット単位での信頼性を保証することを考えており、この点で FEC は他のプロトコルとは異なるため、選択肢から除外している。

これら3種類のプロトコルは特徴が大きく異なり、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。AFDP や RMTP は一般的にファイル転送での使用を想定しており、SRM は分散型マルチメディアアプリケーションを想定して作成された。また、AFDP が規模適応性がないといわれているのに対し、SRM はグループの規模が大きい場合にも対応できるといわれている。

## 3.2 評価

### 3.2.1 評価の方法

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<sup>11)</sup>は非常に多い。本研究では、評価を行う際にシミュレーションを用いた。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。

- (1) 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、たとえば「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。
- (2) 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。
- (3) 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。

(1)においてパケット損失率を設定できるということは、逆にいうと「何が起こるか分からない」という実際のネットワークの状況をつくり出すことはできない。しかし、本稿ではまず各プロトコルの特徴を明らかにすることが目的であり「送信者付近にのみエラーが発生する場合」といった状況を想定することで、各プロトコルが送信者付近のエラーに対してどのような性能を示すかということを知ることができる。

4章では、シミュレーションを用いることによる(1)から(3)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。

### 3.2.2 評価の基準

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フロー制御の効率の良さに注目した。以下に、再送制御、フロー制御の説明と、評価の際にこれらに注目した理由を示す。

#### 3.2.2.1 再送制御

ネットワーク上に何らかの障害が発生し、受信者がすべ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。本稿で注目した3種類のプロトコルは、すべ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、この再送パケットの数が少ないほど、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。

#### 3.2.2.2 フロー制御

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見られた場合には、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が発生することを防ぐために、高信頼性マルチキャストプロトコルの中にはフロー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。フロー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いときには、データを速く配信することができ、状態が悪いときには、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。このスループットが高いほど、効率の良いフロー制御が可能で、性能の良いプロトコルであるとして比較、評価を行う。

## 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

### 4.1 実験環境

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<sup>12)</sup>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、本稿で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。

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)<sup>13)</sup>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想定した。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

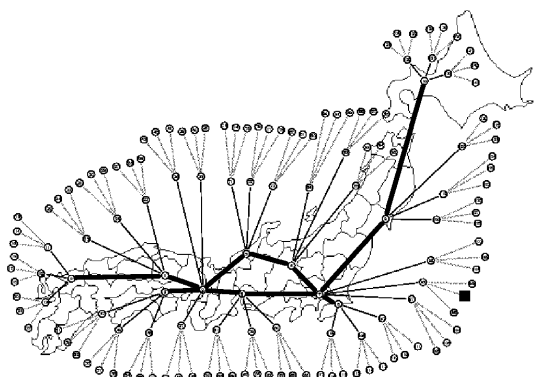


図1 シミュレーションに用いたネットワーク  
(黒(太線): 10 Gbps, 黒(細線): 1 Gbps, 灰色: 100 Mbps, : 送信者)

Fig. 1 Network topology on the simulator  
(black (thick line): 10 Gbps, black (thin line): 1 Gbps, gray: 100 Mbps, : sender).

研究開発施設(ギガビットラボ)を整備し, 大学, 研究機関など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 カ所の ATM 交換設備および 39 カ所の接続装置を超高速光ファイバで結んだ最大 2.4Gbps の帯域を持つ通信回線である。

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 1 に示す。

図 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, 図中の黒い四角の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 が持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, 黒く太い線がバックボーンとなる 10 Gbps の全二重線, 黒く細い線が 1 Gbps の全二重線, 灰色の線が 100 Mbps の全二重線となっている。

#### 4.2 実験(1). 再送制御の評価

まず実験(1)として, 再送制御の評価を行う。実験(1)では, 図 1 に示した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 から 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 カ所のエラー発生リンクを設定し, 各リンクでのパケット損失率は 5% とした。エラーを受信者付近に発生させた場合, 送信者付近で

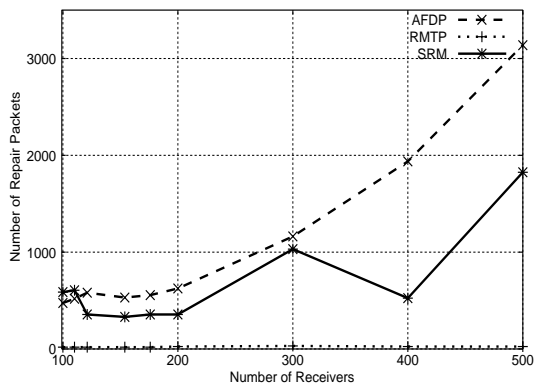


図2 再送パケット数の比較(送信帯域: 10 Mbps)  
Fig. 2 Comparison of the number of repair packets (sender bandwidth: 10 Mbps).

パケットが落ちる場合に比べてデータを正しく受け取れない受信者は限定されるので, 再送要求の発生も少ない。パケット損失率の値については, ルータなどでバースト的にパケットをロスするのではなく, リンクで少しずつパケットをロスする場合を想定しているため, 低めの値に設定した。

送信するデータはパケットサイズが 1,000 バイトであり, 送信帯域が 10 Mbps, 30 Mbps の 2 種類, パケット数は 1,000 個とした。送信帯域については, ファイル配信をアプリケーションとして想定した場合(10 Mbps)と, マルチメディアデータの配信を想定した場合(30 Mbps)の 2 種類について調べた。10 Mbps の方は, 比較的遅くデータが送出されるため, 各リンクへの負荷も少ないが, 30 Mbps の方は負荷が大きいため, 再送制御で多くの再送パケットが流れると, さらにリンクの負荷が高くなってパケット損失率が高くなるという悪循環を生む可能性がある。

まず送信帯域が 10 Mbps の場合の実験結果を図 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, まったく再送パケットを受信しない受信者も存在するため, 1 受信者あたりの平均をとると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

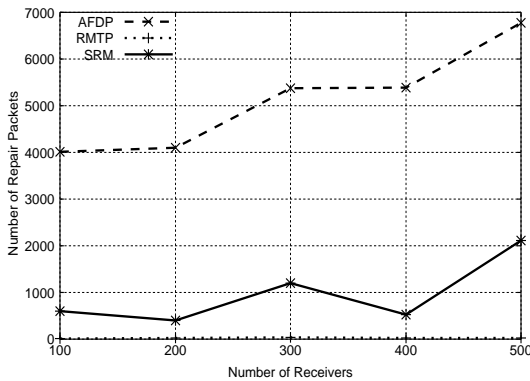


図3 再送パケット数の比較 (送信帯域: 30 Mbps)

Fig. 3 Comparison of the number of repair packets (sender bandwidth: 30 Mbps).

を示している。

次に送信帯域が 30 Mbps の場合の実験結果を図 3 に示す。RMTP の性能は送信帯域が 10 Mbps の場合と変わらず、再送パケット数は非常に少ないのに対し、AFDP はパケット数が非常に多く、グループの規模の大小に関係なく SRM よりも性能が悪いという結果が得られている。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。さらに今回は送信帯域が高いため、ネットワークに負荷がかかり、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った。実際、実験の経過を観察すると、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った。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。

#### 4.3 実験 (2). フロー制御の評価

実験 (2) では、10 メガバイトのファイルを配信し終えるのに要する時間を測定し、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。図 1 に示したネットワークにおいて、100 の受信者に対してデータを配信する。エラーが発生するリンクの場所については、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 2.5% から 10% まで変化させる。送信するデータは、送信帯域 10 Mbps、パケットサイズ 1,000 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という用途を想定している SRM については、データを配信する

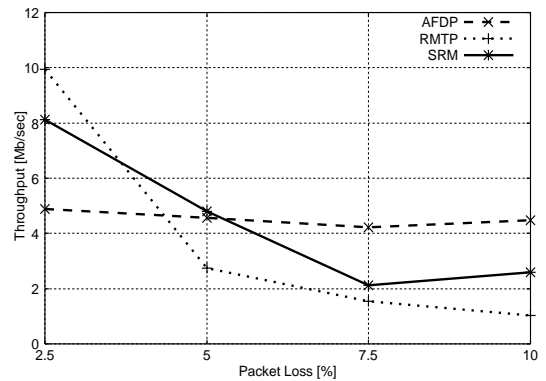


図4 スループットの比較 (受信者付近のエラー)

Fig. 4 Comparison of the throughput (errorlink: near the receiver).

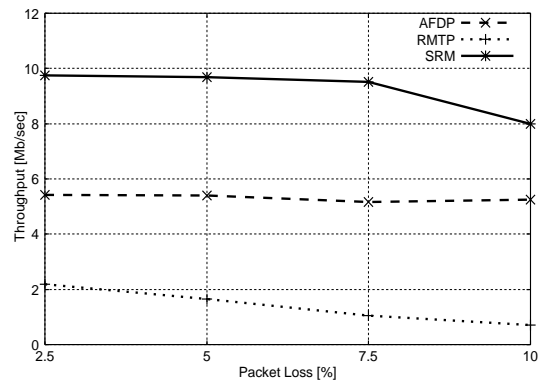


図5 スループットの比較 (送信者付近のエラー)

Fig. 5 Comparison of the throughput (errorlink: near the sender).

際の遅延も少ないと予想される。また、送信者付近でエラーが発生する場合には、各受信者が送信者との間で再送制御を行う AFDP の性能が悪くなるのが予想される。

まず、受信者付近にエラーが発生する場合の実験結果を図 4 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。RMTP と SRM については、パケット損失率が高くなるにつれて、スループットが低下しているのに対し、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ど大きくない。次に、送信者付近にエラーが発生する場合の実験結果を図 5 として示す。この場合、SRM が最も性能が良く、パケット損失率が高くなって性能の低下を生じないという結果が得られている。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 の性能が悪く、AFDP は予想に反して良い性能を示した。これはフロー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 が返ってこ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。

## 5. プロトコル選択方式の評価

### 5.1 実験結果のまとめ

まず、4 章で行った実験の結果をまとめる。図 6、図 7 として、実験結果をまとめた図を示す。図 6 は横軸にグループの規模、縦軸にはデータの送信帯域をとっている。データの種類に関しては、AFDP は送信帯域が低い場合に有効であり、SRM はネットワークに負荷がかかるような、送信帯域の高いデータにおいても、性能がそれほど悪化しないという結果が得られた。しかし、規模適応性を考慮すると、グループの規模が大きい場合には、送信帯域が低い場合においても AFDP より SRM の方が有効であるという結果が得られている。RMTP については、送信帯域、グループの規模に関係なく、再送パケット数に関しては他のプロトコルと比べて最も性能が良いという結果が得られた。今回の実験では、グループの規模は 100 から 500 の間で行ったが、この受信者数の変化の中で、AFDP と SRM の性能は逆転し、RMTP については他の 2 つのプロトコルと比べて明らかに良い性能を示した。500 以上の受信者を持つグループについては、RMTP が最も性能が良く、次に SRM、そして AFDP という傾向が続いていくと考えられる。

図 7 については、横軸にスループット、縦軸にはエラーが発生するリンクの場所をとっている。再送パケット数については、RMTP がエラー発生箇所に関係なく良い性能を示したのに対し、スループットについては良い結果が得られなかった。AFDP は受信者に近い場所でエラーが起きた場合に有効であり、スループットも高い。これに対し、SRM は送信者に近い場所でのエラーに有効であり、高いスループットが得られている。

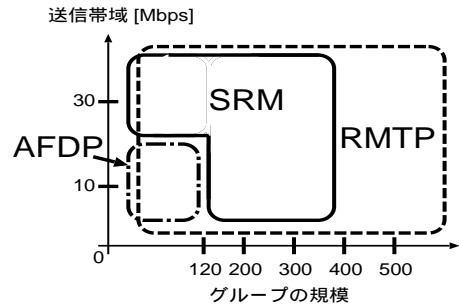


図 6 各プロトコルの適用範囲(再送制御)

Fig. 6 Applicable range of the protocols (retransmission control).

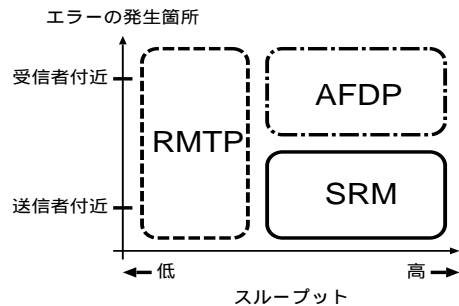


図 7 各プロトコルの適用範囲(フロー制御)

Fig. 7 Applicable range of the protocols (flow control).

今回の実験では、マルチキャストグループの受信者の分布の仕方、すなわち、トポロジによって各プロトコルの性能が大きく異なるという結果が得られた。つまり、同じ受信者数でも、トポロジが変わった場合にはプロトコルの性能が逆転する可能性があるということである。今後、プロトコルを選択するための指標としてシミュレーションによる結果を利用する場合には、トポロジ情報やエラーの発生仕方、アプリケーションの種類などを忠実に再現した状況でのシミュレーションの実行が必要であると考えられる。

### 5.2 複数のプロトコルを選択することによる効果

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の実現には、複数のプロトコルを組み合わせる必要がある。そこで本節では、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が高くな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、4 章の実験で用いたものと同様であり、図 8 のようになる。

4 章での実験と異なるのは、エラーが発生するリンクの場所である。末端に近いところに 3 カ所と、バックボーンのリンクに 1 カ所設定している(パケット損

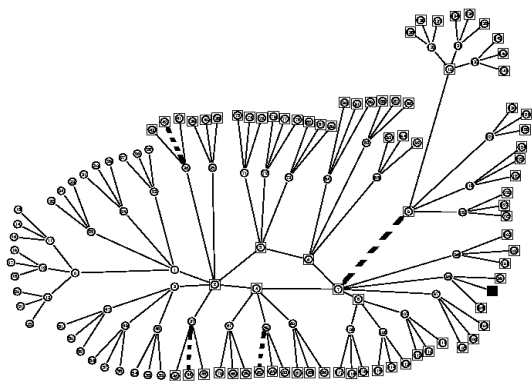


図8 実験環境 (複数のプロトコルを組み合わせさせた場合) (○: 送信者, ●: 受信者, 点線: エラー発生リンク)  
 Fig. 8 Configured network (○: sender, ●: receiver, dotted line: error link).

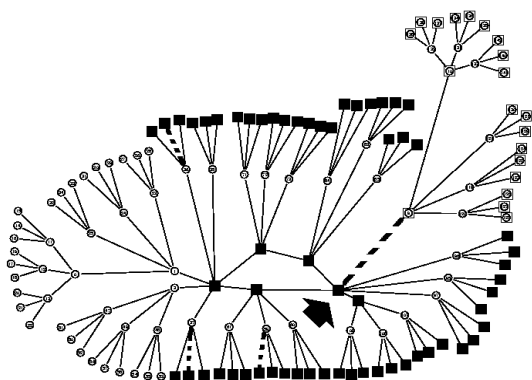


図9 プロトコルの分布状況 (○: RMTP, ●: SRM, 点線: エラー発生リンク)  
 Fig. 9 Distribution of the protocols (○: RMTP, ●: SRM, dotted line: error link).

失率は5%としている)。これにより、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、北日本の受信者の多くがデータを正しく受け取れず、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、図6、図7を参考にし、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9に示す。

図9に示しているように、北日本の受信者に対しては、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、他の場所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リンクの直前のノード(図9において矢印で示したノード)は、RMTPからSRMへのプロトコルの変換を行うゲートウェイとな

表1 複数のプロトコルを組み合わせることによる効果  
 Table 1 Effect of the protocol combination.

	RMTP	SRM	組合せ
再送パケット数(平均)	253	4859	176
スループット [Mbps]	2.49	5.27	4.41

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の送信者として北日本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1,000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1に示す。表1を見ると、再送パケット数については、RMTP, SRMを組み合わせさ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いえる。

## 6. 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、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使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。状況に応じて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いえる。しかし、本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。

今後は、提案している手法の実装を行う。具体的な実装の方針について述べると、提案している手法を実現するには、まず最初に適切なプロトコルを選択するための準備として、送信者がマルチキャストグループの規模を把握する必要がある。送信者は各受信者から受信者レポートを受け取り、この数からグループの規模を把握する。この情報とアプリケーションの種類から送信者は使用するプロトコルを決定する。プロトコルを決定する際には、本稿で示した、シミュレーションの結果を用いる。実際のネットワークをシミュレータ上で実現し、エラー発生率や受信者数に関して数多くの状況を想定したシミュレーションを行い、プロトコルを選択する際の閾値を前もって導出しておくことにより、効果的にプロトコルの選択が行えるようにする。

今後はネットワークの時間的な変化にも対応する予定である。たとえば、データの配信中に受信者の数が大きく変化した場合に、途中で動的にプロトコルを切り替え、それぞれの規模に合ったプロトコルで通信を行うというものである。実現方法としては、データ配信中でも送信者は受信者からのレポートを定期的に獲得し、グループの規模の変化を監視しておく。また、エラー発生への対応は、各受信者からパケット損失に関する情報を集め、パケット損失率の高い箇所については、5.2 節の実験(2)で想定した場合と同じようにある地点から別のプロトコルで送信するという方法で対応する。

提案している手法において考えられる問題点は、まず、データの配信を開始するために必要な準備の時間や、プロトコルを途中で切り替える場合に、その切替えに必要な時間などから発生するオーバーヘッドが考えられる。また、受信者情報の獲得のために、送信者と受信者との間で制御パケットをやりとりする必要があるため、規模適応性を実現するためには受信者の間でパケットの集約を行うなどの対処をする必要がある。あるプロトコルを単独で使用した場合には、これらの処理は必要なく、すぐにデータの配信を開始でき、切替えも必要ない。このようなオーバーヘッドがあることを考えると、複数のプロトコルを使用するよりも、単独のプロトコルで動作した方が高い効率が得られる場合も考えられる。たとえば、配信するファイルのサイズが小さく、配信に要する時間が非常に短い場合には、複数のプロトコルの切替えによるオーバーヘッドにより、スループットが下がってしまう。プロトコル選

択方式を実装する際には、アプリケーションの種類によって、単独のプロトコルで通信するという選択肢を含めておくことも必要であろう。

今後はこのプロトコル選択方式の実装を行い、実際のネットワークを使った実験を行って、先に述べたような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。また、この方式を適用できるアプリケーションの種類を増やすために、今回選択した3種類のプロトコル以外のプロトコルや、FECなどの手法を選択肢として含めていくことも考慮したい。

謝辞 本稿を執筆するにあたり、様々なご助言をいただいた WIDE プロジェクトの RM ワーキンググループの皆様、九州大学の荒木研究室、岡村研究室の皆様に、謹んで感謝の意を表す。また、本研究の一部は日本学術振興会未来開拓学術研究推進事業 (JSPS-RFTF 99P01301) の支援を受けている。

## 参考文献

- 1) Miller, C.K.: *Multicast Networking and Applications*, Addison-Wesley (1998).
- 2) Cooperstock, J. and Kotsopoulos, S.: Why Use a Fishing Line When You Have a Net? An Adaptive Multicast Data Distribution Protocol, *Usenix '96* (1996).
- 3) 園田雅崇, 岡村耕二, 荒木啓二郎: 汎用性のある高信頼性マルチキャストに関する研究, 情報処理学会マルチメディア, 分散, 協調とモバイル (DICOMO 2000) シンポジウム論文集, pp.223-228 (2000).
- 4) Sonoda, M., Okamura, K. and Araki, K.: Study of General Reliable Multicast by Use of Active Network Framework, *Proc.15th INTERNATIONAL CONFERENCE ON INFORMATION NETWORKING*, pp.825-830 (2001).
- 5) Whetten, B., Vicisano, L., Kermode, R., Handley, M., Floyd, S. and Luby, M.: Reliable Multicast Transport Building Blocks for One-to-Many Bulk-Data Transfer, RFC 3048 (2001).
- 6) 米山清二郎, 砂原秀樹: FEC on IPv6 for Reliable Multicast, インターネットコンファレンス 2000 論文集, pp.65-74 (2000).
- 7) 田島佳武, 森川博之, 青山友紀: リライアブルマルチキャストのための分散キャッシングの効果に関する検討, 情報処理学会マルチメディア, 分散, 協調とモバイル (DICOMO '99) シンポジウム論文集, pp.13-18 (1999).
- 8) Paul, S., Sabnani, K.K., Lin, J.C.-H. and Bhattacharyya, S.: RMTP: A Reliable Multicast Transport Protocol, *IEEE Journal on Selected Areas in Communications*, Vol.15,

pp.407–421 (1997).

- 9) Floyd, S., Jacobson, V., Liu, C., McCanne, S. and Zhang, L.: A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing, *1995 ACM SIGCOMM Conference*, pp.342–356 (1995).
- 10) Stevens, W.: TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms, RFC 2001 (1997).
- 11) Mankin, A., Romanow, A., Bradner, S. and Paxson, V.: IETF Criteria for Evaluating Reliable Multicast Transport and Application Protocols, RFC 2357 (1998).
- 12) ns2 (network simulator version2): <http://www.isi.edu/nsnam/ns/>
- 13) JGN (Japan Gigabit Network): <http://www.jgn.tao.go.jp/>

(平成 13 年 6 月 6 日受付)

(平成 13 年 10 月 16 日採録)



柴田 賢介

1977 年生。2001 年九州大学工学部電気情報工学科卒業。同年九州大学大学院システム情報科学府情報工学専攻入学。インターネット上における高信頼性マルチキャスト通信に

関する研究に従事。Wide Project 会員。



岡村 耕二 (正会員)

1965 年生。1990 年九州大学大学院工学研究科情報工学専攻修了。三菱電機株式会社情報電子研究所入所、奈良先端科学技術大学院大学情報科学研究科助手、神戸大学総合情報処理センター助手、九州大学情報処理教育センター助教授、2000 年九州大学情報基盤センター助教授、現在に至る。博士(工学)。次世代インターネットインフラストラクチャ、高度マルチメディアデータ通信の研究開発に従事。教育システム情報学会、ソフトウェア科学会、Wide Project 会員。日本学術振興会 163 委員会(インターネット技術研究委員会, ITRC) 委員。



荒木啓二郎 (正会員)

福岡市生まれ。1976 年九州大学工学部卒業、1978 年同大学大学院修士課程修了。九州大学助手、同助教授、奈良先端科学技術大学院大学教授を経て、現在、九州大学大学院システム情報科学研究院教授(財)九州システム情報技術研究所研究室長兼務。工学博士。形式仕様記述、ソフトウェア開発方法論、プログラミング言語、並列/分散処理、インターネット、マルチメディア通信等の研究に従事。日本ソフトウェア科学会、ACM、IEEE CS 等の会員。ソフトウェア技術者協会常任幹事、ふくおかギガビット委員会座長、博多祇園山笠西流元赤手拭い等。