

# NACKに基づいた単純な高信頼マルチキャスト伝送方式の提案

M2004MM030 中本 拓也

指導教員 長谷川 利治

## 1 はじめに

近年一対多, 多対多の通信の需要が高まり multicast を用いた通信が注目されている. しかし既存の multicast は UDP を使って通信を行うため信頼性が無い. よって現在 RM(Reliable Multicast) が研究されている.

配信データの信頼性を確保する方法の中から, NACKに基づいた損失パケット再送により信頼性を確保する, NORM(Negative-Acknowledgment(NACK)-Oriented Reliable Multicast)[1] を使った通信に着目する. 既存の NORM の仕様は RFC3940 で定義されているが, 仕組みが複雑であることやまだ実験段階であることから改善すべき点がある.

本研究の目的は, NORM の仕組みを簡素化しより使いやすく, また柔軟性を持たせた信頼性のあるマルチキャスト, SNMLR(Simple NACK-based Multicast with Local Recovery) を提案することである. NORM と本研究で扱う SNMLR の違いは, 損失パケット再送をローカルノード(ホスト)により行う点, FEC を用いない点, NACK 抑制に用いられるタイマー間隔を短くした点, 必要でないメッセージをすべて省いた点である. これらにより, 既存の NORM よりも簡単な仕組みで無駄なトラフィックを抑え, かつ同程度の性能を得ることが可能となる.

本研究では NORM と SNMLR を性能評価した. 実験は, NORM や SNMLR を組み込んだホストをシステムのエンド部分に設置し, 小規模ネットワークを構築する. そして実ネットワークに NORM または SNMLR を用いてファイルを転送し, 実験結果より SNMLR の有効性を示す.

## 2 これまでの RM の進展

RM に対する要求はアプリケーションごとに色々であり, すべての AP 要求を統一的に満足させる唯一の RM を設計することは不可能とされている. そのため様々な目的に応じた RM が提案されているが, ネットワーク制約があったり, 仕組みが複雑であったり, 問題点はさまざまである.

### 2.1 RM に求められる多様な要求

RM プロトコルを設計する際には, アプリケーションの種類や目的に応じて配送に求められる要求が異なり, 表 1 のように分類することができる. その中でも特に実時間性と信頼性いずれかを求められることが多いが, 株価配信といった実時間性も信頼性も必要とするシビアなアプリケーションにも対応できる RM も求められる. 各アプリケーション要件により適応可能な RM も変化する.

表 1 アプリケーションの要求

	信頼性	スケーラビリティ	実時間性	セキュリティ
POSレジデータ	◎	◎	×	○
工場間CADデータ	◎	×	×	○
オンラインゲーム	×	○	◎	×
株価配信	◎	○	○	×
ビデオ会議	×	×	◎	×

### 2.2 データの信頼性保証

信頼性を確保する方法として, 損失パケットを再送することによる回復と, そのデータに対する冗長データを配信することによる回復がある.

#### 損失パケット再送による回復

これは, 受信者がパケット損失を検知し再送要求し, 送信者から該当パケットを再送してもらうことでパケット補完を可能とする方法である. ACK ベース, NACK ベース, ACK 集約ベースなどによって再送要求を行う方法があるが, 受信者数によっては Feedback implosion が起きてしまうため, 工夫が必要である.

#### 冗長パケットによる回復

データを配信するだけでなく, 前方誤り訂正 (FEC) を用いて冗長パケットを配信したり, 同一パケットを複数回連続して送信することで, 損失パケットの再送を抑えることができる. よってネットワークが混雑している時などに有効と言える.

FEC を用いる手法としては proactive と reactive の 2 通りがある. proactive はあらかじめ通常データに加えて冗長データを配信し, reactive は受信者からのパケット要求がきたら冗長データを配信する. proactive 型は受信者からの再送要求を減らすことができ, reactive 型ではネットワークの状況によって冗長データによるトラフィックを減らすことができる.

### 2.3 プロトコルの標準化

現在 IETF(Internet Engineering Task Force) の rmtwg(Reliable multicast transport working group) では 1 つのプロトコルに絞るのではなく, 以下に示す 2 つのプロトコルの標準化が進められている [2].

1. NACK ベースプロトコル
2. FEC を利用した非対称階層符号化プロトコル (ALC)

1 は NACK を用いて再送要求を出すプロトコルであり, 多量のデータ欠落が発生しない限り NACK implosion が発生しにくく, 無駄なトラフィックを減らすことがで

きる。2 は FEC を用いて受信側で損失パケットを補完することができ、損失があっても回復できるので NACK 再送を減らすことができる。

### 2.4 主な NACK を用いた RM と SNMLR の違い

本研究で提案する SNMLR は、ローカルリカバリとグローバルリカバリを組み合わせ、ローカルリカバリで回復できないものをグローバルリカバリにより回復を図る。これによりネットワークにあまり負荷を与えることなく、早期のパケット補完が可能と考えられる。また、ローカルリカバリの前身として White Board がある [3]。

過去に研究されている SRM[4] と同様に、タイマにより NACK を抑制する [5]。違いは、SRM では一度近隣ノードから再送パケットをもらうローカルリカバリ [6] を図り、失敗した場合送信ノードへ再送を求めるグローバルリカバリを図るが、SNMLR では近隣ノードへ何度も再送を求め、それでも無理ならグローバルリカバリを図るという点である。SNMLR はそのさい少しずつ NACK を送る範囲を広げていくが、SRM は近隣ノードへ NACK を送るさいもグループ全員へ届けられ局所性がない。SRM では他にマルチキャストグループを分けて小さなグループを作り、近いグループのところへ NACK を送り再送要求をする手法もある。しかし両者ともまだ実装されていない。

## 3 単純な高信頼マルチキャスト伝送方式の提案

この節では、本研究で扱う SNMLR の仕組みやベースとなる NORM について、NACK 再送の仕組みについて説明する。

### 3.1 NORM について

NORM による信頼性は、受信者からの NACK に基づき送信者からパケットを再送することで保証される。再送要求はパケット損失発見後タイマを起動し、タイムアウト後に送信元までマルチキャストされる。受け取った送信元は該当パケットをマルチキャストで再送し、パケット補完を可能とする。

また、NORM データは UDP パケットにカプセル化されて送信されるので、既存の multicast 対応ルータで扱うことができる。

### 3.2 SNMLR による信頼性保証

本研究で提案する SNMLR は、NORM を単純化し、損失パケットをローカルリカバリで回復するプロトコルである。その目的は、パケット再送の高速化とネットワーク全体の再送トラフィックの軽減、損失パケットの早期回復である。そのためパケット再送要求を近くのノードに送ることで、該当パケットを近くのノードが持っていたら、そのノードからパケットを送信しローカルリカバリを図る。それにより、NACK implosion の抑

制や早期データ回復を実現することができる。

なお、FEC を用いたローカルリカバリは、誤り訂正のための情報を受信ノードが持っていないとできないため、proactive 型でのみ実装可能である。NORM の example コードでは、誤り訂正のための情報は送信元しか持っていない reactive 型だけが含まれていたため、実験ではなしとした。

### 3.3 ローカルリカバリを用いた再送方式の提案

マルチキャスト通信において、ノードがパケットを正しく受信できない原因はいろいろ考えられるが、SNMLR が特に効果を発揮するのは、

- 受信ノードの処理が遅い
- 送信元から受信ノードまでのコストがかかる

などである。

SNMLR における再送要求方法は以下の通りであり、NACK が配送される範囲の例を次ページ図 1 に示した。

[ローカル再送要求]

パケット損失を発見したノード (n1) が、local-backoff timer を起動して微小時間待ち、その後 ttl 2 で NACK を送信 (マルチキャスト) する。それを受け取ったノードが該当するパケットを正しく受信できていれば、同じ ttl で該当パケットを再送しローカルリカバリを図る。n1 が NACK を送信した後 local-backoff timer を再び起動し、タイムアウトするまでに repair パケットがなければ ttl を増やし NACK を送信し、さらにローカルリカバリを図る。それでも該当パケットが届かなければ、n1 は global-backoff timer を起動し、タイムアウト後送信者へ NACK を送信 (マルチキャスト) してグローバルリカバリを図る。

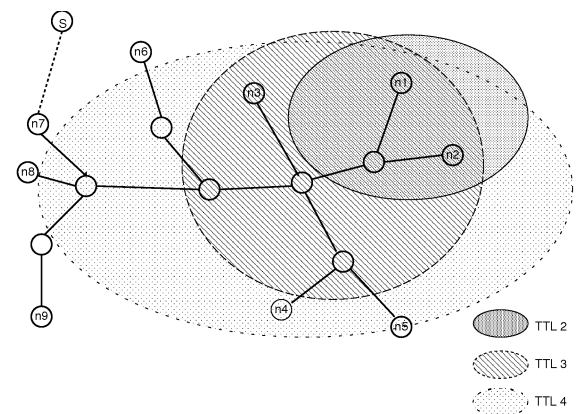


図 1 NACK の有効範囲

図 1 は、送信者 (S) から作られる木構造のマルチキャストネットワークにおいて、ノード n1 がパケット損失をし NACK を送り出す有効範囲を示したものである。NACK を受信可能なノードは ttl 2 : n2

ttl 3 : n2, n3

ttl 4 : n2, n3, n4, n5

となるのがわかる。このように repair パケットが必要な受信ノードに近いところからローカルリカバリを求めることで、すばやいパケット補完を可能とする。ttl を 2 から始めるのは、1 だとルータを越えないからである。

## 4 評価

この節では SNMLR の有効性を評価する。シミュレーションと実ネットワークにおいてファイル転送を行い、NORM と SNMLR の性能を評価する。実験には、開発段階ではあるが、文献 [7] を公開している NORM プロジェクトチームにより無償提供されている NORM プログラムを使う。SNMLR には、NORM プログラムに変更を加え SNMLR を実現するプログラムを作り、それをを用いる。

### 4.1 NORM の再送システムの評価

パケットロス障害のある実験ネットワークで NORM を用いてマルチキャストを行い、実験した結果を表 2 にまとめた。NORM は FEC をサポートしているため、FEC を用いたときのデータである。このときのネットワークは同一サブネット上で、受信ノード 1 台、送信ノード 1 台を用意した。よって送信ノードと受信ノードのコストは 0 なので、このデータは NORM に実装された再送システムの評価となる。

表 2 パケットロス時のデータ転送完了時間

パケットロス確率	元データ送信 パケット数 (再送を除く)	NACK送信 パケット数	再送パケット数 (server-> receiver)	総送信パケット数 (server-> receiver)	データ転送 完了時間
0%	38	0	0	38	1.715 sec
5%	38	0.8	1.6	39.6	6.684 sec
10%	38	1.6	5	43	9.714 sec

NORM での問題点を以下に示す。

- NACK 抑制のためのタイムが長く、パケット回復まで時間がかかる
- NACK はすべてのノードに送られるため、無駄なトラフィックが増える (ユニキャスト送信も可能だが他ノードは NACK を受け取らないので NACK 抑制ができず implosion を引き起こしやすい)

SNMLR は、ローカルリカバリによる損失パケット回復を可能とし、これらの問題点を解決することができる。ローカルリカバリのための NACK は、NORM のタイムよりも短い間隔で設定されているため、より早くパケット回復を実現できる。送信される NACK は NORM より多いが、近隣ホストにのみ送られるのでネットワークに大きな負荷をかけることなく送信できると考えられる。

### 4.2 実ネットワークでの実験

マルチキャスト通信の性能評価には、そのスケーラビリティを計るために多数のノードを配置したネットワークを用意する必要がある。しかし数百~数千ものノードを用意するのは現実的ではない。そこでシステム全体をエミュレートするのではなく、エンドシステム部だけの小規模なネットワークを用意し、SNMLR の実ネットワークでの有効性を計る。図 2 は本研究で用いるネットワークモデルである。

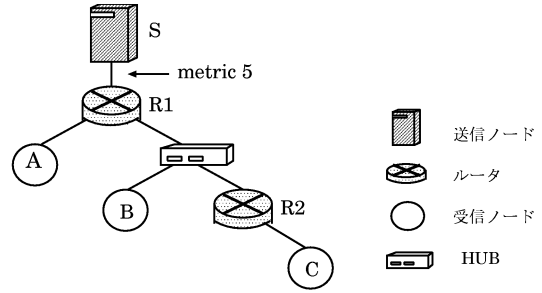


図 2 実ネットワークのモデル

送信ノード S から次のルータ R1 までの距離が離れていることを実現するために、metric を大きくした。そのためローカルリカバリ中の NACK や再送パケットはノード S へ届くことはない。

実際に与えた各パラメータは以下の通りである。ロスモデルとは、各ノードで起きるパケット損失率である。また、評価対象として、NORM, SNMLR の他に、FEC を用いない NORM も加えた。

転送レート : 10Mbps

ファイルサイズ :  $2 \times 10^6$  Byte

ロスモデル : S:10% A:5% B:15% C:10%

遅延 : S  $\leftrightarrow$  R1 10ms, R1  $\leftrightarrow$  A 5ms, R1  $\leftrightarrow$  HUB 5ms

HUB  $\leftrightarrow$  B 5ms, HUB  $\leftrightarrow$  R2 5ms, R2  $\leftrightarrow$  C 10ms

#### 4.2.1 各ノードのトラフィックによる評価

各ノードにおけるトラフィックを比較した。結果を表 3 に示す。

表 3 各ノードのトラフィック

	IN			OUT		
	NORM	no-FEC	SNMLR	NORM	no-FEC	SNMLR
ノードS	7.1	13	5.8	3319	4455	2445
ノードA	2529	2709	2532	1.5	2.9	714
ノードB	2310	2610	2654	3.2	5.6	551
ノードC	2440	2586	2601	4.3	13	412

NORM より SNMLR の方が、ノード S より出されるトラフィックを約 25% 抑えることができた。これはローカルリカバリにより、損失パケット再送が大きく減ったためである。また、ノード A, B, C において

repair パケットを送信するために、OUT トラフィックが大きく増えた。

#### 4.2.2 ノード S における NACK, repair 数の比較

ノード S における NACK 数, repair 数を比較した。結果を表 4 に示す。

表 4 NACK, repair 数の比較

	NACK			Repair		
	NORM	no-FEC	SNMLR	NORM	no-FEC	SNMLR
ノード S	18	19	11	1025	1743	342

再送パケットによるネットワーク消費を約 65% 抑えることができ、大きく SNMLR の優位性が見られた。受信ノードが増えた場合でも送信ノードへの影響をあまり与えることなくデータ転送ができ、スケーラビリティの高い通信が可能である。

NACK の数があまり減ってはいないがこれは NACK 数と repair 数の関係を見てもわかるように、NACK は一つの NACK だけで複数個のデータ要求が可能である。SNMLR では NORM とは違い、一つの NACK で要求する repair 数は少なく、送信ノードへ送られるトラフィックは十分少なくなった。

#### 4.2.3 転送完了時間による評価

次にファイル転送完了時間による評価をする。転送完了条件は、ノード S において転送開始時から、送信するデータがなくなったときまでとする。ここでは転送レートを 100kbps, 1Mbps, 10Mbps, ファイルサイズを  $100 \times 10^3$ ,  $2 \times 10^6$  Byte で実験を行ったときのデータを比較対象とした。単位は sec(秒) である。

表 5 転送完了時間による比較

転送レート	$100 \times 10^3$			$2 \times 10^6$		
	NORM	no-FEC	SNMLR	NORM	no-FEC	SNMLR
100kbps	14.4	16.4	18.9	214.5	229.8	205.3
1Mbps	8.1	9.3	8.2	31.2	26.4	27.9
10Mbps	5.4	10.6	10.4	76.4	30.5	28.2

表 5 から、ファイルサイズが大きいものに対して SNMLR の有効性が見られたが、小さいものに対してはあまり時間の短縮にはならなかった。ファイルが小さい場合すぐオリジナルデータを送り終えてしまうので、それ以後はグローバルリカバリによる回復が主となり時間がかかる。グローバルリカバリによる再送システムは NORM と同じであるため転送完了時間の短縮にはあまりつながらなかった。時間を短縮するためには、グローバルリカバリによる再送システムで使われる global-backoff timer, holdoff timer for recv の再検討が必要である。

## 5 おわりに

RM を実現するに当たり、NACK ベースのマルチキャストではスケーラビリティに大きな優位点をもっており、今後さらに研究されていく分野である。NACK ベースのマルチキャストが抱える問題点として NACK 信頼性問題や TCP 親和性問題があるが、ローカルリカバリの範囲を制御することでこれらの問題を解決した。

本研究では、NORM と SNMLR を用いた実験を行った。SNMLR を使うことで、NORM より送信ノードへの負担を大きく減らすことができた。また転送完了時間もサイズの大きいファイルでは NORM より短縮することができた。しかし NACK 抑制タイムがボトルネックとなり、ファイルサイズによってはあまり期待した効果を得ることができなかった。

今後の課題は、シミュレーションツールを用いて大規模ネットワークにおける SNMLR 有効性を計ることである。また、各タイムリカバリできる範囲を再検討し転送完了までの時間短縮を計り、ストリーミングデータ配信における性能を評価する必要がある。そして他の既存 RM との性能比較も必要である。

## 参考文献

- [1] Brian, A., Carsten, B., Mark, H., and Joe, M.: "Negative-acknowledgment (NACK)-Oriented Reliable Multicast (NORM) Protocol," RFC3940(2004).
- [2] "Reliable Multicast Transport (rmt) Charter," <http://www.ietf.org/html.charters/rmt-charter.html>(参照 2005-9).
- [3] Floyd, S., Jacobson, V., Liu, C., McCanne, S., and Zhang, L.: "A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing," IEEE/ACM Transactions on Networking, vol.5, No.2, pp.784-803, Dec (1997).
- [4] 村本衛一: "実用的な高信頼性マルチキャストに関する研究," 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 (2000).
- [5] Atwood, J.W.: "A Classification of Reliable Multicast Protocols," IEEE Network Mag., vol.18, pp.24-35, May/June (2004).
- [6] Liu, C., Estrin, D., Shenker, S., and Zhang, L.: "Local Error Recovery in SRM: Comparison of Two Approaches," IEEE Transactions on Networking, vol.6, pp.686-699, Dec (1998).
- [7] The Naval Research Laboratory (NRL) PROTOcol Engineering Advanced Networking (PR OTEAN) Research Group.: "NORM - NACK-Oriented Reliable Multicast," <http://pf.itd.nrl.navy.mil/norm/>(参照 2006-1).