高信頼性 XCAST プロトコルの設計

伊藤政利† 下川俊彦†† 山下雅史††

インターネットや広帯域ネットワークの普及により，サーバのミラーリングが重要になってきた．本論文では，その通信に IP マルチキャストの代わりに XCAST を用いることを提案する．IP マルチキャストはグループ数に対する規模適応性がないため，サーバの数だけグループ数を必要とするミラーリングに用いることは不適切である．そこで，グループ数に対する規模適応性がある XCAST を用いる．本論文では XCAST の通信を高信頼化した「高信頼性 XCAST プロトコル」を開発した．このプロトコルは送信者が送信した全パケットを全受信者が必ず受け取ることができるプロトコルである．このプロトコルにより，ミラーリングの際に生ずるトラフィック量やパケット数の軽減を実現する．本論文では，設計した高信頼性 XCAST プロトコルについて説明する．そして，シミュレーションによる評価を行い，このプロトコルの有用性を述べる．

Construction for Reliable XCAST Protocol

Masatoshi Ito,† Toshihiko Shimokawa†† and Masafumi Yamashita††

Deployment of the Internet and broad-band networks increase the importance of mirrors of servers. In this paper, we propose to use XCAST, instead of IP multicast, to implement mirrors, since IP multicast is not scalable in the number of multicast groups and hence is inappropriate for a mirroring protocol, which needs to manipulate the number of multicast groups the same as the number of servers; XCAST, on the other hand, is considered to be scalable in the number of multicast groups. To reduce the number of packet exchanges and then to reduce the amount of traffic concerning the mirroring, this paper proposes a protocol called Reliable XCAST, which is an extension of XCAST so that it guarantees reliable communication. After explaining Reliable XCAST, this paper conducts simulations to demonstrate the usefulness of Reliable XCAST.

1. はじめに

近年，インターネットや広帯域ネットワークの普及により，CDN（Contents Distribution Network）[1]が注目されている．CDN におけるミラーサーバは複数用意するのが一般的である．したがって，ミラーリングを行う際の通信方法はユニキャストを複数回行うよりも，マルチキャストを 1 回行う方が有効となる．

インターネットで一般的なマルチキャストとしては IP マルチキャストがある．しかし，IP マルチキャストにはグループ数に対する規模適応性がない．ミラーリングにおいて，通常マスターサーバからのパケットを受信する複数のミラーサーバが 1 つのグループを形成するので，マスターサーバの数だけグループが存在することになる．したがって，IP マルチキャストにグループ数に対する規模適応性がないことが致命的な問題となる．

この問題を解決するマルチキャストとして XCAST（Explicit Multicast）が提案されている[2)~4)]．XCAST はグループ数に対する規模適応性があるマルチキャストである．本研究では CDN のミラーリングに着目し，XCAST を用いてミラーリングのためのファイル配信プロトコルを設計した．

† 九州大学大学院システム情報科学府
Graduate School of Information Science and Electrical Engineering, Kyushu University
†† 九州大学大学院システム情報科学研究院
Faculty of Information Science and Electrical Engineering, Kyushu University

2. XCAST（Explicit Multicast）

本章では，IP マルチキャストにおける問題点と，新しいマルチキャスト技術である XCAST（Explicit Multicast）について述べる．

2.1 IP マルチキャスト

インターネットで用いられている一般的なマルチ

キャストとしては IP マルチキャストがある．IP マルチキャストでは，グループをマルチキャストアドレスで示す．このアドレスにはインターネット内において一意性が必要である☆．

ホストがグループに参加すると，このグループを示すマルチキャストアドレス宛のパケットを受信可能になる．この際，パケットの配送はルータが行う．受信者のグループにマルチキャストアドレスを割り当てた後，配送木上にあるすべてのルータに 1 つ 1 つのアドレスに対する経路情報を設定する必要がある．これにより，マルチキャスト経路表がグループ数の増加に比例して大きくなり，グループ数に対する規模適応性がない[5]．

これはミラーリングにおいて致命的な問題である．IP マルチキャストを用いてコンテンツを配信するためには，マスターサーバからのパケットを受信するすべてのミラーサーバで 1 つのグループを形成する必要がある．このグループはマスターサーバの数だけ存在することになる．したがって，グループ数に対する規模適応性がない IP マルチキャストを用いるのは不適切である．そこで，別のマルチキャストを用いなければならない．

2.2 XCAST

IP マルチキャストの問題を解決するマルチキャストとして XCAST（Explicit Multicast）が考え出された．XCAST では，送信したい複数の宛先のユニキャストアドレスすべてをパケットに明示して送信する．つまり，グループをマルチキャストアドレスではなく，宛先のユニキャストアドレスのリストで表す．マルチキャストアドレスを用いず，ユニキャストアドレスしか用いないため，グループ数が増えても経路表が大きくなることはない．したがって，グループ数に対する規模適応性は高い．以降，宛先ユニキャストアドレスのリストを宛先リストと呼ぶ．

「複数の宛先を明示して送信する」という XCAST の概念を用いたプロトコルには，MDO6[2]や SGM[3]，Xcast[4]等がある．本研究では Xcast を検討対象とする．

3. 高信頼性 XCAST

本章では，まず本研究で開発した高信頼性 XCAST プロトコルを説明する．次に，これを用いたファイル配信プロトコルについて説明する．

☆ 技術的にはパケットが届く範囲を限定することも可能である．この場合はパケットが届く範囲で一意性が必要となる．

3.1 XCAST の高信頼化

XCAST の通信は UDP を用いているため，通信の途中でパケットが紛失する可能性がある．届かないパケットが 1 つでも存在すると，パケットからファイルを復元できなくなってしまう．そこで，XCAST の高信頼化を行う．

高信頼化を行うには再送機能と再送要求機能が必要となる．これらの機能について説明する．

3.1.1 再送機能

再送を行うノードとしては送信者，途中のルータ，他の受信者の 3 つがある．ここでは再送を行うノードを再送者と呼ぶことにする．

再送する方法は次の 3 つを考えることができる

(1) ユニキャスト再送
(2) IP マルチキャスト再送
(3) XCAST 再送

このそれぞれについて説明する．

(1) ユニキャスト再送
再送を要求してきた各受信者にユニキャストで再送を行う方法である．したがって，再送が必要な受信者にだけ送信することができる．しかし，同じパケットを再送する場合，要求した受信者の数だけ再送を繰り返さなくてはならない．

(2) IP マルチキャスト再送
グループ全体に IP マルチキャストで再送を行う方法である．したがって，同じパケットの再送を要求している受信者が複数存在した場合でも，再送者は 1 回の再送で済む．しかし，再送を必要としない受信者にもパケットを再送してしまう．

(3) XCAST 再送
同じパケットの再送を要求してきた受信者に XCAST で再送を行う方法である．したがって，同じパケットの再送を要求している受信者が複数存在した場合でも，再送者は 1 回の再送で済む．また，再送を必要としている受信者にしかパケットを再送しない．

3.1.2 再送要求機能

再送を要求する方法には大きく次の 3 つの方法がある．

(1) ACK（ACKnowledgement）
(2) NACK（Negative ACKnowledgement）
(3) SACK（Selective ACKnowledgement）

それぞれについて**図 1** を用いて説明する．図 1 における a，b，c はそれぞれパケットの一連番号であり，図 1 は b のパケットが紛失していることを示して

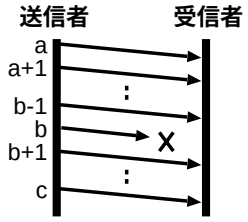

図 1 再送要求
Fig. 1 Request for retransmission.

いる．

(1) ACK
「受け取った」情報を返す方法である．たとえば，図 1 の場合では「b − 1 までのパケットを受け取った」という ACK を送信者に返す．
しかし，送信者がこの ACK を「b のパケットが届いていない」と解釈するには，たとえば同じ ACK を複数回受け取ることが必要になる．

(2) NACK
「届いていない」情報を返す方法である．たとえば，「b のパケットが届いていない」という NACK を送信者に返す．この NACK を受け取った送信者は，b のパケットを再送すればよい．しかし，この NACK からだけでは，その受信者はこの一連番号以外のパケットを受け取っているのか確認できない．したがって，NACK だけでは送信者はいつまでも送信した内容を覚えておかなければならない．

(3) SACK
「受け取った」情報と「届いていない」情報の両方を返す方法である．たとえば，「a から b − 1 までと b + 1 から c までのパケットを受け取った」という SACK を送信者に返す．この SACK を受け取った送信者は，b のパケットを再送すればよい．また，受け取ったパケットも分かるので，いつまでも送信内容を覚えておく必要もない．

3.2 本研究で設計した高信頼性 XCAST プロトコルの概要

本節では，設計した高信頼性 XCAST プロトコルの概要を説明し，次節以降で送信者と受信者に分けて詳しく説明する．

本研究で設計した高信頼性 XCAST プロトコルの処理の流れの概要は以下のようになっている．

(1) XCAST 送信
本研究ではミラーリングを対象としているので，グループ数に対する規模適応性が必要である．そのため，XCAST でパケットを送信する．

(2) SACK 送信
3.1.2 項で述べた 3 つの再送要求方法の中で SACK が最も効率的であると考え，再送要求には SACK を用いる．受け取ったパケットと受け取っていないパケットの両方の情報を載せるために，受信者はある程度パケットを受け取った後に SACK を送信する．
受信者に届くパケットは，Xcast の X2U 機能によってユニキャストパケットとして届く．このため，他の受信者に関する情報を知ることはできない．また，途中のルータが再送をするとなると，新しいルータを導入しなければならなくなる．そこで，本プロトコル受信者は送信者に SACK を送信する．

(3) XCAST 再送
理想的な再送とは「再送を要求した受信者だけにマルチキャストで再送」することである．これを，3.1.1 項で述べた再送方法について検討する．ユニキャスト再送では明らかに理想を満たすことは無理である．IP マルチキャスト再送では，動的にグループを形成することは困難であるため，理想的な再送をすることは困難である．XCAST 再送では，宛先リストを変更するだけで，新たなグループを形成できる．したがって，再送パケットごとにグループを作成してマルチキャストすることが可能であり，理想的な再送を行うことができる．そこで，本プロトコルでは XCAST による再送を用いる．
送信者は再送要求を受け取ってもすぐには再送を行わず，しばらくの間，他の受信者からの再送要求を待つ．こうして，複数の受信者に対して，XCAST での送信が可能になる．

3.3 送信者側の動作

送信者のパケットの送信は大きく 2 つに分けることができる．1 回目のパケット送信と，そのパケットが紛失した場合の紛失パケットの再送である．1 回目のパケットのことを「本パケット」と呼び，本パケット送信のことを，再送に対して「本送信」と呼ぶことにする．この 2 つの送信について説明する．

3.3.1 本 送 信

本送信において，送信者は全受信者にパケットを XCAST で送信する．その際，受信者は紛失していないか調べなければならない．そのため，送信者は送信するパケットには一連番号をつける．

3.3.2 再　　送

送信者は受信者から SACK を受け取り，紛失があれば，そのパケットを再送をしなければならない．この際，送信者は XCAST で再送をする．ただし，送信者は SACK を受け取ってもすぐには再送を開始せずに，しばらく待った後で再送を開始する．これは，SACK を受け取ってすぐに再送を開始すると，後で他の受信者から同じパケットの再送要求を受けた場合，もう一度そのパケットを再送をしなければならなくなってしまうからである．この，再送をしばらく待つ時間のことを，SACK 収集時間と呼ぶことにし，その時間を Tcs とする．

送信者が再送を開始する契機は以下の 2 つである．

- SACK 収集時間の経過
  SACK を受信した送信者は，紛失パケットごとに SACK 収集のタイマを起動させる．Tcs の間に他の受信者からの SACK を受け取って紛失を検出した場合，このパケットがすでに再送予定のパケットであれば，その再送パケットの宛先リストに後から受け取った受信者のアドレスを追加する．紛失したパケットが再送予定でなければ，新規に再送パケットを用意して Tcs 後に再送する．
- 全受信者からの SACK 受信
  受信者は SACK を送信すると，基本的には，この SACK で要求したパケットを受信してから次の SACK を送信する（3.4 節参照）．したがって，全受信者から SACK を受け取るということは，全受信者が再送パケットを待っている状態であり，送信者が再送するまで受信者は SACK を送信しない．そこで，送信者は全受信者から SACK を受け取ると，Tcs 経過しなくても再送を開始するようにした．

3.3.3 受信者管理

XCAST では，送信者はだれにパケットを送信しているのか完全に把握できる．しかし，送信者はその受信者が本当に存在しているかは分からない．そこで本プロトコルでは，受信者からの SACK を生存確認にも利用することにした．まず，一定時間（Tws）SACK を送信してこない受信者がいると，この受信者に SACK_REQ メッセージを送り，SACK を送信するように要求する．さらに Tws 時間経過しても SACK が来なければ，再び SACK を要求する．これを 5 回繰り返しても SACK が来なければ，送信者はこの受信者に何らかの障害が起きたと見なし，この受信者を宛先リストから外してパケットを送信しないようにする．

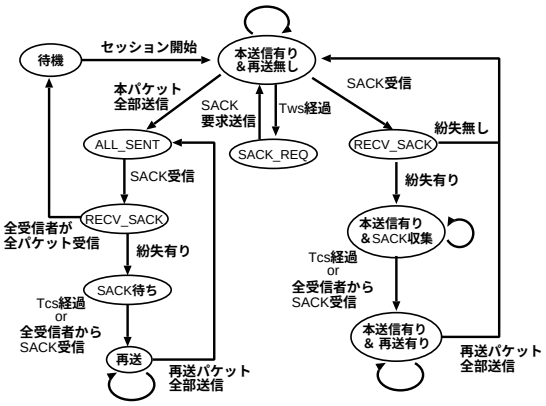

図 2　送信者側の状態遷移図
Fig. 2　The state transition diagram for sender.

3.3.4 送信者側の状態遷移図

送信者側の処理についてまとめると，**図 2** のような状態遷移図となる．

3.4 受信者側の動作

受信者は紛失があれば，SACK を送信して再送要求しなければならない．なお，3.3.2 項で述べたように，送信者は紛失パケットごとに SACK 収集のタイマを起動させる．したがって，受信者が頻繁に SACK を送信すると，送信者はタイマをいくつも起動するために負荷がかかってしまう．そこで，受信者は SACK を送信すると，この SACK で要求した再送パケットが到着するまで，できる限り次の SACK を送信しないようにした．

以上より，SACK を送信する契機は「再送パケット待ちではない状態」「再送パケット待ち状態」「状態に無関係」の大きく 3 つの場合に分けて考えなければならない．それぞれの場合について考察し，受信者が SACK を送信する契機としては以下の 8 通りとした．

(1) 再送パケット待ちではない状態
  (a) 前の SACK 送信から一定パケット数の受信
  (b) 前の SACK 送信から一定時間の経過
  (c) 最後のパケットの受信
  (d) 前の本パケット受信から一定時間の経過
(2) 再送パケット待ちの状態
  (a) 再送パケットを全部受信
  (b) 再送パケットの紛失
  (c) 再送要求から一定時間の経過
(3) 状態に無関係
  (a) SACK_REQ メッセージ受信

このそれぞれについて説明する．

3.4.1 再送パケット待ちではない状態の SACK 送信

- 前の SACK 送信から一定パケット数の受信
  生存確認のため，受信者はある程度の頻度で SACK を送信する必要がある．しかし，あまり頻繁に SACK を送信するとトラフィック量が増えてしまう．そこで，以前に SACK を送信したときから一定パケット数（N パケット）を受け取ると SACK を送信するようにした．
- 前の SACK 送信から一定時間の経過
  前の SACK 送信から N パケット受け取ると SACK を送信するようにした．しかし，N パケットを受け取って SACK を送信するまでの時間は受信者によって大きく異なる場合もある．この場合，送信者が SACK を受信するタイミングに差が生じてしまい，ある受信者からの SACK を受け取ってから Tcs 時間が経過しても，他の受信者からの SACK をわずかしか収集できなくなってしまう．そこで，以前に SACK を送信したときからの一定時間（Tss）が経過しても SACK を送信するようにした．
- 最後のパケットの受信
  再送パケット待ちではない状態で最後のパケットを受け取ると，SACK を送信することにした．Tss 時間待てば SACK を送信するが，3.3.2 項でも述べたとおり，送信者は全受信者からの SACK を受け取ると再送を開始する．そこで，最後のパケットを受け取ったときに SACK を送信することで送信者が他の受信者へ早く再送開始できる．
- 前の本パケット受信から一定時間の経過
  インターネットではパケットが急にまったく届かなくなるという場合もある．そこで，一定時間（Tnp）が経過しても次のパケットが届かなければ SACK を送信するようにした．これも Tss 時間待っても SACK を送信することになるが，最後のパケットを受信したことによる SACK 送信のときと同じ理由により，このようにした．

3.4.2 再送パケット待ちの状態の SACK 送信

本項では，再送パケット待ち状態における SACK 送信の契機について述べる．なお，再送パケットを待っている間にも本パケットは送信されてくる．この本パケットも紛失する可能性がある．再送パケット待ち状態での SACK 送信には，この分の再送要求もその SACK の中に含めることにした．

- 再送パケットを全部受信
  SACK で要求したパケットが全部届いたら，再送パケットが届いたことを報告するため SACK を送信する．
- 再送パケットの紛失
  再送も UDP を用いているので要求したパケットが全部届くとは限らない．そこで，一部の再送パケットを受け取ったが，一定時間（Tnp）経過しても次の再送パケットが届かなければ，SACK を送信して再びそのパケットを要求する．なお，Tnp は「前の本パケット受信から一定時間の経過」における Tnp と同じ値とする．
- 再送要求から一定時間の経過
  再送を要求したが，いくら待っても再送パケットがまったく届かない場合もある．そこで，再送パケットが届かないまま一定時間（Twr）経過すると SACK を送信するようにした．

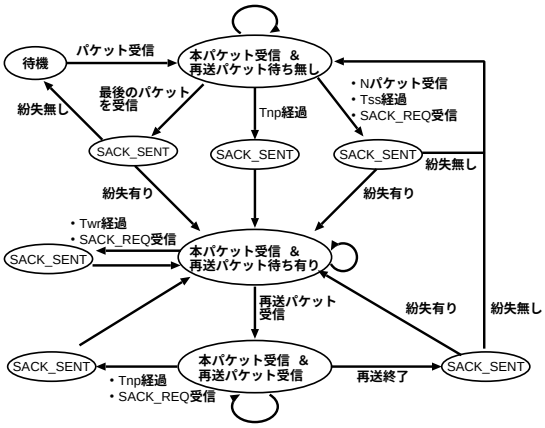

図 3 受信者側の状態遷移図
Fig. 3 The state transition diagram for receiver.

3.4.3 状態に無関係の SACK 送信

- SACK_REQ メッセージ受信
  3.3.3 項で述べたように，本プロトコルでは受信者からの SACK で受信者の生存を管理している．送信者に SACK が届かないまま Tws 時間経過すると，送信者は SACK_REQ メッセージを送信する．したがって，このメッセージを受信した受信者の SACK は送信者に届いていないことを示すので，状態に関係なく SACK を送信する．

3.4.4 受信者側の状態遷移図

受信者側の処理についてまとめると，**図 3** のような状態遷移図となる．

3.5 ファイル配信プロトコルの目標

ミラーリングでは大量のファイルを複数の受信者に配信することになる．したがって，ミラーリングという応用を考えた場合，トラフィック量が重要となる．

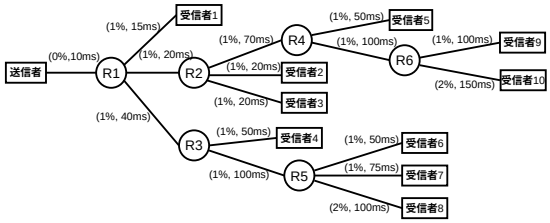

図 4 3.6 節と 4 章で行うシミュレーションに用いる評価用ネットワーク
Fig. 4 The network used in simulations in Sections 3.6 and 4.

そこで，本プロトコルの設計においてはトラフィック量を少なくすることを目標とした．

3.6 SACK 収集時間の決定

予備実験の結果，トラフィック量を減らすには SACK 収集時間が重要な要素であることが分かった．そこで本節では，最適な SACK 収集時間の値を決定するために行ったシミュレーションについて述べる．

3.6.1 シミュレータへの組み込み

3.2 節から 3.4 節で設計した高信頼性 XCAST プロトコルをネットワークシミュレータ ns-2 [7]に組み込んだ．その際，実装を簡単にするためにネットワークトポロジは静的なものとし，配送木が変化しないものとした．

3.6.2 シミュレーションネットワークの構築

シミュレーションに用いたネットワークを**図 4** に示す．丸や四角で囲んでいるのが送信者や受信者，ルータ等のノードであり，それらを結ぶ線がリンクである．丸で囲まれた R1 から R6 はそれぞれ XCAST 対応ルータを表す．各リンクに付いている括弧の中の 2 つの値は，それぞれ紛失率と遅延を表す．ネットワークトポロジやパケット紛失率，送信者からの遅延は，各受信者が様々な状態となるように設定した．このネットワークを用いて最適な SACK 収集時間を決定した．

3.6.3 準　　備

本プロトコルにおける時間やパケット数に関するパラメータには以下のものがある．

- SACK 収集時間（Tcs）
- 次のパケットまでの待ち時間（Tnp）
- 再送パケット待ち時間（Twr）
- SACK 送信間隔（Tss）
- SACK 待ち時間（Tws）
- SACK を送信するまでの受信パケット数（N）

最適な SACK 収集時間を求めるため，Tcs の値を変化させてシミュレーションを行うことにした．その他のパラメータ値は以下のとおりとする．

- Tnp：0.2 [s]
- Twr：3.0 [s]
- Tss：3.0 [s]
- Tws：6.0 [s]
- N：1024 [パケット]

最適な Tcs の値を調べるため，以下の項目を測定する．

(1) XCAST 再送の場合の理想再送回数
以下の仮定をした場合の送信者の再送回数である．
  - 送信者は同じ再送パケットを複数回送信しない．
  - 再送を要求した全受信者は必ず再送パケットを受け取る．

(2) ユニキャスト再送の場合の理想再送回数
再送要求を受けたパケットをユニキャストで再送し，かつ再送パケットが紛失しないと仮定した場合の送信者の再送回数である．これは，各受信者における紛失パケット数の総和と同値である．

(3) 送信者の実際の再送回数
送信者が実際に行った再送回数である．これには再送パケットの紛失による再送も含まれる．

(4) XCAST 再送効率
ユニキャストではなく XCAST で再送することにより，どれくらい効率が良くなったかを表す．これは，(2) の値に対する (3) の値の割合である．

(5) 受信者の送信回数
受信者が SACK を送信した回数である．

(6) 本送信以外のパケット数
本送信以外に送信者と受信者が送信したパケット数である．これは，(3) と (4) の和である．

各値は 10 回の平均値をとることにした．

3.6.4 測　　定

図 4 のネットワークにおいて，512 bytes のデータを載せたパケットを 10,000 パケット送信し，3.6.3 項で述べた項目を測定した．その結果を**表 1** に示す．

XCAST 再送の場合の理想再送回数とユニキャスト再送の場合の理想再送回数はパケットの紛失率を固定に設定しているので，有意な変化はない．

SACK 収集時間が長くなると，送信者の実際の再送回数は少なくなっていき，途中からはほとんど変わらなくなる．SACK 収集時間が長いとそれだけ多くの受信者からの SACK を集めることができる．これによって，1 回の XCAST でより多くの受信者に再送できるため，実際の再送回数が減る．しかし，受信者の

表 1 異なる SACK 収集時間における測定結果
Table 1 Simulation results for some SACK-collection time values.

SACK 収集時間 [s]	XCAST再送の場合の理想再送回数	ユニキャスト再送の場合の理想再送回数	送信者の実際の再送回数	XCAST 再送効率	受信者の送信回数	本送信以外のパケット数
0.80	1560	2859	2228	0.78	83	2311
0.85	1583	2889	2185	0.76	80	2265
0.90	1560	2839	1780	0.63	74	1854
0.95	1579	2823	1770	0.62	73	1843
1.00	1562	2849	1688	0.59	69	1757
1.15	1564	2850	1685	0.59	70	1755

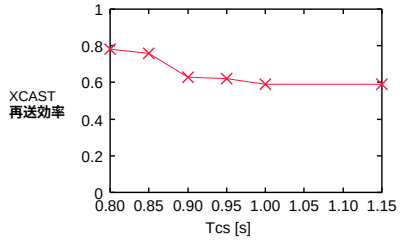

図 5 SACK 収集時間と XCAST 再送効率との関係
Fig. 5 The relationship between SACK-collection-time and XCAST-retransmission-efficiency.

数以上には SACK は来ないので，途中からこの項目の値はほとんど変わらなくなる．

SACK 収集時間が長くなると，XCAST 再送効率は小さくなり，途中からはほとんど変わらない．これはユニキャスト再送の場合の理想再送回数に対する，実際の再送回数の割合であることから明らかである．SACK 収集時間が長くなると，受信者の送信回数は少なくなり，途中からはほとんど変わらない．これは SACK 収集時間が長くなれば，それだけ SACK 送信回数が減るためである．

SACK 収集時間が長くなると，本送信以外のパケット数は少なくなり，途中からはほとんど変わらない．これは，送信者の再送回数と受信者の送信回数の和なので明らかである．

XCAST 再送効率の値に着目し，Tcs の値と XCAST 再送効率の値をグラフ化したものを**図 5** に示す．図 5 を見ると明らかなように，XCAST 再送効率は Tcs が大きくなるにつれて減少し，Tcs = 1.00 [s] で落ち着いている．一方，Tcs を大きくすると再送パケットが届くまでの時間が長くなる．SACK が紛失しなければ，送信者は全受信者からの SACK が届いた時点で再送を開始するため，Tcs が大きくても問題とはならない．しかし，ある受信者の SACK が紛失した場合，送信者は SACK が紛失したことを知ることができない．したがって，Tcs 時間待った後で再送を開始しなければならない．そのため，受信者が再送パケットを

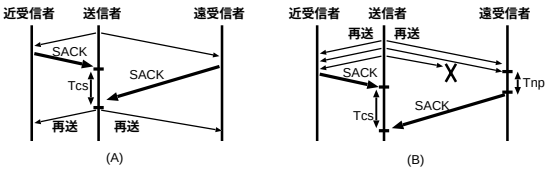

図 6 SACK 収集時間，最大 RTT と最小 RTT の差，そして Tnp の間の関係
Fig. 6 The relationship among the SACK-collection time, the difference between the maximal and minimal RTT's and Tnp.

受け取るのが遅くなってしまう．したがって，図 4 のネットワークの場合は，Tcs の値は 1.00 がよいことになる．

このように，Tcs の値はある時点で最適となる．このときの Tcs の値を最適 Tcs と定義する．次に，最適 Tcs の値の求め方について考える．

3.6.5 最適 Tcs の求め方

図 4 のネットワークだけでなく，どのようなネットワークにおいても最適 Tcs の値があると予測した．その最適 Tcs の値は，以下の理由から全受信者中の最大 RTT と最小 RTT の差と，Tnp の値の 2 つに関係があると考えた．

たとえば**図 6** の (A) のように，送信者が送信したパケットを受け取って，受信者が SACK を送信する場合を考える．送信者がパケットを送信してから最も早く受け取る SACK は，最も RTT が小さい受信者の SACK であり，これに要する時間はおよそこの受信者への RTT である．最初の SACK を受け取った送信者は他の受信者からの SACK を待つ．そして最も遅く受け取る SACK は，最も RTT が大きい受信者の SACK であり，これに要する時間もこの受信者への RTT である．したがって，送信者は最大 RTT と最小 RTT の差の時間待っていれば，全受信者からの SACK を集めることができることになる．

また，本プロトコルにおいて受信者は再送パケットを待っている間はタイムアウトにならない限り SACK を送信しない．このため，最も頻度が多い SACK 送信の契機は，再送パケットを全部受信したことによる SACK 送信，または再送パケットの紛失による Tnp 経過後の SACK 送信となる．前者は，紛失率の小さい受信者，つまり送信者に近い受信者である確率が高い．逆に後者は紛失率の大きい受信者，つまり送信者に遠い受信者である確率が高い．よって，最適 Tcs は図 6 の (B) ように，RTT の差と Tnp の和の関数と考えることができる．

そこで，この考えが正しいことを確認するため，図 4

表 2 RTT の差を変化させた場合の最適 Tcs（Tnp = 0.5）
Table 2 The optimum Tcs's for some RTT's (Tnp = 0.5).

		最大RTTと最小RTTの差 [s]			
		0.5	0.65	0.7	0.9
最大RTT [s]	0.7	1.20	1.30		
	1.1	1.15	1.40	1.35	1.60
	1.5	1.20	1.30	1.35	1.55

のネットワークにおいて最大 RTT，最小 RTT を変化させて，なおかつ Tnp の値を 0.5 s に変えてそれぞれの場合において最適 Tcs の値を測定した．この結果を**表 2** に示す．表 2 を見て分かるとおり，各受信者の RTT を変えても最大 RTT と最小 RTT の差が変わらなければ，最適 Tcs の値もほとんど変わらない．そして RTT の差が 0.65 のところを見ると，Tnp の値を大きくした分だけ，最適 Tcs の値も大きくなっている．

したがって，最適 Tcs は最大 RTT と最小 RTT の差と，Tnp の関数であるということができる．そして，最適 Tcs は以下の式で求めることができる．

$$\text{最適 Tcs} = \text{最大 RTT と最小 RTT の差} + \text{Tnp} + \alpha \quad (\alpha = 0.35 \sim 0.45) \cdots\cdots (1)$$

また，α は処理時間となる．

以上より，いかなるネットワークにおいても，そのネットワークの最大 RTT と最小 RTT の差を知ることができれば，トラフィック量を少なくする最適 Tcs の値を式 (1) で決定できることが分かった．この式 (1) を用いて，本プロトコルの最適 Tcs を決定する．

4. 評　　価

本プロトコルの性能を評価するため，シミュレーションにより異なるプロトコルとの比較を行った．

本論文では NACK を用いた場合と比較する．しかし，3.1.2 項で述べたように，NACK だけでは送信者は終了を確認できない．そこで，全パケット受け取った場合だけは ACK を用いることにした．

NACK と ACK を用いたプロトコルを以下のように設計した．

(1) 受信者は紛失を検出すると紛失パケットごとに NACK を送信者に送信する．
(2) NACK を受け取った送信者はすぐに再送せずに，他の受信者からの NACK を待つ（NACK 収集時間）．
(3) 送信者は NACK 収集時間経過すると，そのパケットを XCAST で再送する．
(4) 受信者は Twr 時間経過しても再送パケットが来なければ，NACK を再送する．

表 3 NACK+ACK との比較
Table 3 A comparison between SACK and NACK+ACK.

プロトコル	XCAST再送の場合の理想再送回数	ユニキャスト再送の場合の理想再送回数	送信者の実際の再送回数	XCAST 再送効率	受信者の送信回数	本送信以外のパケット数
NACK + ACK	1555	2798	1621	0.58	2902	4523
	1575	2819	1631	0.58	2918	4549
	1584	2886	1656	0.57	2995	4651
	1552	2868	1632	0.57	2989	4621
	1573	2858	1646	0.58	2961	4607
	1568	2855	1643	0.58	2963	4606
本プロトコル	1562	2849	1688	0.59	69	1757

表 4 NACK+ACK とのトラフィック量の比較
Table 4 A comparison of traffic between SACK and NACK+ACK at low packet loss rates.

プロトコル	本パケット（再送パケット）の大きさ [bytes]	再送要求パケットの大きさ [bytes]	NACK (SACK) 収集時間 [s]	送信者の実際の再送回数	受信者の送信回数	本送信以外のトラフィック量 [bytes]
NACK + ACK	776	52	0.80	1621	2902	1,408,800
			0.85	1631	2918	1,417,392
			0.90	1656	2995	1,440,796
			0.95	1632	2989	1,421,860
			1.00	1646	2961	1,431,268
			1.05	1643	2963	1,429,044
本プロトコル	776	560	1.00	1688	69	1,348,528

(5) 全部のパケットを受け取った受信者は送信者に ACK を返す．

つまり，本プロトコルの SACK に関する部分を NACK と ACK に変更したプロトコルである．以降，これを NACK+ACK プロトコルと呼ぶ．このプロトコルを図 4 のネットワークに組み込んで，NACK 収集時間を変えながら本プロトコルの場合と同じ項目を測定した．**表 3** がその結果である．

表 3 を見ると，受信者が実際に送信した回数も NACK+ACK プロトコルの方が多くなっている．これはパケットを紛失するたびに NACK を送信しているからである．したがって，その値と送信者が実際に再送した回数の和である本送信以外のパケット数の値も多くなる．

トラフィック量を比較する．まず各パケットの大きさを計算する．本プロトコルは IPv6 を対象としている．Xcast はつねに 10 の受信者の宛先リストを持っていると考え，Xcast ヘッダは 216 bytes と仮定する．また，一連番号を 4 bytes とし，SACK は連続して受け取ったパケットの初めと終わりの一連番号を，つねに 16 組格納しているとして 512 bytes とする．これらの値を用いて，各パケットの大きさと本送信以外のトラフィック量を計算した結果が**表 4** である．

表 4 を見て分かるとおり，わずかな差ではあるが，本プロトコルの方がトラフィック量が少ない．したがっ

表 5 紛失率を小さくした場合のトラフィック量の比較
Table 5 Comparison of traffic SACK and NACK+ACK in low loss rate.

プロトコル	本パケット(再送パケット)の大きさ [bytes]	再送要求パケットの大きさ [bytes]	NACK (SACK) 収集時間 [S]	送信者の実際の再送回数	受信者の送信回数	本送信以外のトラフィック量 [bytes]
NACK + ACK	776	52	0.80	173	313	150,524
			0.85	169	298	146,640
			0.90	180	314	156,008
			0.95	179	321	155,596
			1.00	165	291	143,172
			1.05	175	311	151,972
本プロトコル	776	560	1.00	177	76	179,912

て，トラフィック量で考えても本プロトコルの方が効率的であるといえる．

また，NACK+ACK プロトコルにおける受信者の送信回数は，パケット紛失数 +1 である．もし，パケット紛失率が小さいならば，受信者の送信回数が減るため，本送信以外のトラフィック量が本プロトコルよりも少なくなる可能性がある．そこで，ネットワークの紛失率を小さくして同じシミュレーションをした．元々の紛失率は図 4 に書いたとおりである．この各リンクにおける紛失率をそれぞれ 10 分の 1 にして，本プロトコルと NACK+ACK プロトコルのトラフィック量を測定することにした．その結果を**表 5** に示す．

表 5 を見て分かるとおり，紛失率が小さいと，本プロトコルよりも NACK+ACK プロトコルを用いた方がトラフィック量が少ない．したがって，パケット紛失率が小さいネットワークにおいては，本プロトコルは NACK+ACK プロトコルに比べてパケット数は少ないが，トラフィック量は多くなることが分かった．

以上のことから，NACK+ACK プロトコルと比較すると，本プロトコルは次のことがいえる．

- 利点
  - 本送信以外のパケット数が少ない．
  - パケット紛失率が大きいと NACK+ACK よりもトラフィック量が少なくなる．
- 欠点
  - 再送要求パケットの大きさが大きい．
  - パケット紛失率が小さいと NACK+ACK よりもトラフィック量が多くなる．

したがって，本プロトコルはパケット紛失率が大きければ効果的であることが分かった．

5. おわりに

5.1 まとめ

インターネットや広帯域ネットワークの普及により，CDN が注目されている．CDN におけるサーバのミラーリングには，信頼性のあるマルチキャストが有効である．

そこで，本研究では XCAST の通信の信頼性を高めた，高信頼性 XCAST プロトコルを設計した．通信の信頼性を高めるには，再送機能と再送要求機能が必要である．理想的な再送とは，再送を要求した受信者だけにマルチキャストで再送することである．そこで本研究では，この理想的な再送を実現するために，再送にも XCAST を用いた．この再送機能で，送信者の送信パケット数の減少とトラフィック量の減少を実現した．再送要求には SACK を用いることにした．この再送要求機能で，受信者の送信パケット数の減少とトラフィック量の減少を実現した．これらの再送機能と再送要求機能を用いて高信頼性 XCAST プロトコルを設計した．

本研究ではさらに，今回設計した高信頼性 XCAST プロトコルを基に，トラフィック量を少なくすることを目標としたファイル配信プロトコルを設計した．トラフィック量を少なくするには，送信者が SACK を受け取ってから再送を開始するまでの時間が重要であることが分かった．この時間を SACK 収集時間（Tcs）と定義した．そして，最適な Tcs の値を決定するためシミュレーションを行った．

その結果，最適 Tcs は以下の式で決定できることが分かった．

$$\text{最適 Tcs} = \text{最大 RTT と最小 RTT の差} + \text{Tnp} + \alpha \quad (\alpha = 0.15 \sim 0.25) \cdots\cdots (1)$$

Tcs を式 (1) で求めた値に設定し，シミュレーションで他のプロトコルとの比較を行った．本論文では，SACK を用いずに NACK+ACK を用いたプロトコルと比較した．その結果，パケット紛失率が小さいネットワークの場合は，本プロトコルの方が NACK+ACK プロトコルよりもトラフィック量が多くなってしまうことが分かった．

5.2 今後の課題

本研究における今後の課題として，まず，最適 Tcs の動的設定がある．今回はネットワークを変化させていない．しかし，現実のインターネット上の実環境ではネットワークトポロジは刻々と変化するため，送信者からの遅延や RTT も変化する．したがって，3.6.5 項で求めた式 (1) における「最大 RTT と最小 RTT の差」も変化するため，通信の途中で最適 Tcs が変わってしまう．そこで，ネットワークの変化に対応して，動的に最適 Tcs に設定する仕組みを考える必要がある．これにより，本プロトコルはインターネット上の実環境でもつねに最適 Tcs で動作する．また，このこ

とをインターネット上の実環境で実際に動かして確認しなければならない．

さらに，再送要求の動的変更がある．ネットワークの状況に従って，動的に再送要求を NACK+ACK，または SACK に切り替えることができれば，つねにトラフィック量の少ない再送要求方法での動作が可能になる．このようなプロトコル設計を行いたい．

また，Tcs 以外のパラメータについても最適な値を決定しなければならない．その際，まずはシミュレーションによって決定する必要がある．そして，最適 Tcs と同様，決定した値についてインターネット上の実環境において確認をしなければならない．

謝辞 日頃の研究活動において様々な助言をいただきました研究室の皆様，また XCAST に関して議論の場と最新の情報を提供していただきました WIDE Project の XCAST Working Group の皆様に深く感謝いたします．

参 考 文 献

1) Day, M., et al.: A Model for Content Internetworking (CDI), Internet Draft, draft-day-cdnp-model-08.txt (Oct. 2001).
2) Imai, Y.: Multiple Destination Option on IPv6 (MDO6), Internet Draft, draft-imai-mdo6-02.txt (Mar. 2000).
3) Boivie, R. and Feldman, N.: Small Group Multicast, Internet Draft, draft-boivie-sgm-02.txt (Feb. 2001).
4) Ooms, D., et al.: Explicit Multicast (Xcast) Basic Specification, Internet Draft, draft-ooms-xcast-basic-spec-02.txt (Oct. 2001).
5) Sola, M., Ohta, M. and Maeno, T.: Scalability of Internet Multicast Protocols, *Proc. INET'98* (July 1998). http://web.jet.es/sola/inet98.html
6) Cidon, I., Khamisy, A. and Sidi, M.: Analysis of Packet Loss Processes in High Speed Networks, *IEEE Trans. Inf. Theory*, Vol.IT-39, No.1, pp.98–108 (1993).
7) The Network Simulator — ns-2. http://www.isi.edu/nsnam/ns/

(平成 14 年 3 月 28 日受付)
(平成 14 年 6 月 4 日採録)

伊藤 政利

2000 年九州大学工学部電気情報工学科卒業．2002 年同大学大学院システム情報科学府情報工学専攻修士課程修了．

下川 俊彦（正会員）

1990 年九州大学工学部情報工学科卒業．1992 年同大学大学院工学研究科情報工学専攻修士課程修了．同年（株）東芝入社．1997 年九州大学大学院システム情報科学研究科情報工学専攻助手．2000 年同大学院システム情報科学研究院情報工学部門助手．博士（情報科学）．広域分散協調処理，インターネット等の研究に従事．

山下 雅史（正会員）

1974 年京都大学工学部情報工学科卒業．1977 年同大学大学院工学研究科情報工学専攻修士課程修了．1980 年名古屋大学大学院工学研究科博士課程後期情報工学専攻修了．同年豊橋技術科学大学教務職員．1981 年同大学助手．1985 年広島大学工学部助教授．1992 年同教授．1998 年以降九州大学大学院システム情報科学研究院情報工学部門教授．博士（工学）．並列/分散アルゴリズム，並列/分散計算論等の研究に従事．