

## マルチホーム端末における 高速ファイル転送方式の提案

高橋大斗<sup>†</sup> 中村嘉隆<sup>††</sup> 白石 陽<sup>††</sup> 高橋 修<sup>††</sup>

モバイル端末の高性能化に伴い、扱うファイルのサイズは肥大化し続けている。本研究では、アドホック無線通信によるファイル転送を高速化することで、制約が少なく処理が速い、つまりモバイル端末間での手軽なファイル交換手法として活用できると考える。ファイル転送高速化の手法として、本研究では CMT と GridFTP-APT という2つの技術に注目した。マルチホームを有効活用する CMT と、帯域を有効活用する GridFTP-APT は非常に親和性が高いと考えられるためである。本研究ではこれらの技術を組み合わせ、マルチホーム端末同士のアドホック無線通信における、高速なファイル転送方式を提案する。また、アドホック無線通信というある種限定された環境下において、CMT と GridFTP-APT という2つの技術の有効性を、実機を用いた実験より評価する。

### Proposal of high-speed file transfer in a multihomed terminal

HIROTO TAKAHASHI<sup>†</sup>  
YOSHITAKA NAKAMURA<sup>††</sup> YOH SHIRAISHI<sup>††</sup>  
OSAMU TAKAHASHI<sup>††</sup>

High-performance mobile terminals are increasing. With this, file size handled by the mobile terminals is growing. This research is considered to exchange files between mobile terminals, in an ad-hoc wireless networks that can be leveraged. This research focused on CMT and GridFTP-APT for speeding up. CMT is the effective use of multihoming. And GridFTP-APT is the effective use of bandwidth. We propose an efficient file transfer method for multihomed terminals in ad-hoc wireless networks, using both the CMT and the GridFTP-APT. We also show the implementation of our proposal method, and performance evaluation of it.

<sup>†</sup>公立はこだて未来大学大学院

Graduate School of Systems Information Science, Future University Hakodate

<sup>††</sup>公立はこだて未来大学

Future University Hakodate

## 1. はじめに

近年、隣接したモバイル端末間において、ファイル交換を行う機会が増えている。ファイル交換の際に着目される点は手軽さであり、制約の少なさ、いかに短時間で交換が行えるかが主題となる。まず、第一の着目点である、ファイル交換時の制約について考える。隣接したモバイル端末間でのファイル交換の手法としては、メール添付による交換、外部記憶デバイスを用いた交換など様々に考えられるが、転送可能なファイルサイズに上限が設けられている場合がある、記憶デバイスを携帯しておく必要があるなど、いずれも使用する際には制約が発生してしまう。隣接したモバイル端末間における、最も制約の少ないファイル交換手法は、アドホック無線通信を用いたファイル転送であると考えられる。端末同士で直接に行うアドホック無線通信の場合、ファイルサイズの上限などの問題は発生せず、また近年のモバイル端末においては、無線ネットワークインタフェースは必ずと言ってほど搭載されているためである。さらに、近年の無線ネットワークインタフェースの普及・低価格化により、一台の端末に複数の無線ネットワークインタフェースを持つ、マルチホーム端末も一般化して来ている。

次に、第二の着目点である、短時間でのファイル交換について考える。近年のモバイル端末の高性能化に伴い、扱うファイルのサイズも肥大化している。数十 MB ~ 数 GB のファイルをアドホック無線通信で転送する機会も増えており、従来のファイル転送手法では時間がかかりすぎる場合があると考えられるため、ファイル転送の高速化は重視すべき点である。ファイル転送高速化の手法として、本研究では Concurrent Multipath Transfer (CMT) [1] と GridFTP-APT [2] に注目した。マルチホーム端末同士が結ぶ複数の経路を同時に使い、帯域幅の拡張を実現する CMT と、コネクション数の調整を行い、帯域を有効活用する GridFTP-APT のアルゴリズムは、親和性のある技術と考えたためである。

本研究ではこれらの技術を参考に、マルチホーム端末同士のアドホック無線通信における、高速なファイル転送方式を提案する。

## 2. 関連研究

この章では、本研究で注目する並列転送技術である CMT と、コネクション数調整アルゴリズムである GridFTP-APT について述べる。

### 2.1 Concurrent Multipath Transfer (CMT)

マルチホーム端末間において、ネットワークデバイスごとに複数の経路を結ぶことを、マルチパスと呼ぶ。Concurrent Multipath Transfer (CMT) とは、一つの大きなデータを分割し、マルチパスによる複数の経路を同時に利用して転送する技術であり、同時に使用する経路のぶん、単位時間当たりの転送量を増加させることが可

能となる[1]. シングルパスによる従来の転送方式と、パス2本で行ったCMTによる転送方式を、図1で比較している。

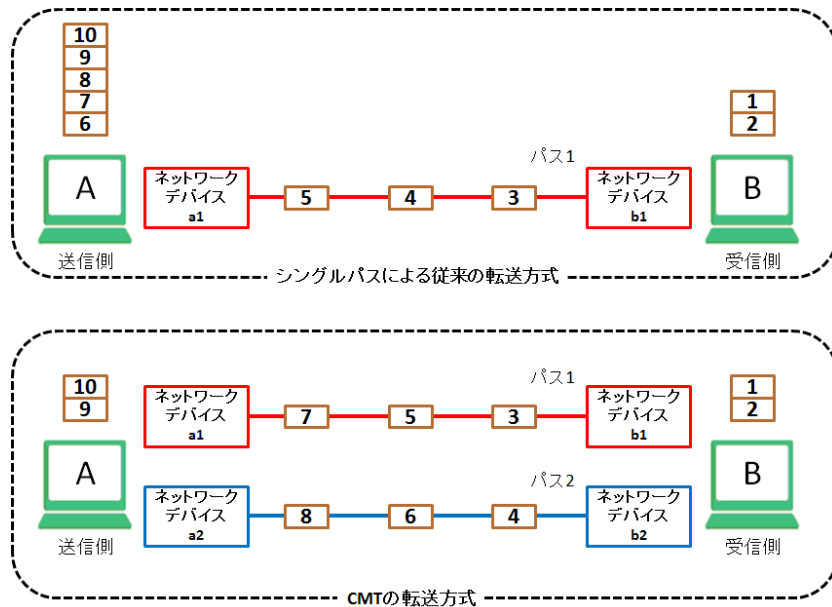


図1 転送手法の比較

## 2.2 GridFTP-APT

GridFTP-APT[2]とは、GridFTP[3]におけるTCP接続数の最適値を求めるアルゴリズムである。GridFTPとは、グリッドコンピューティング向けに開発されたファイル転送プロトコルであり、転送対象のファイルを断片化し、TCP接続毎に並列転送を行って帯域を有効活用することで、高速なファイル転送を可能としている[4]。また、ファイルを断片化して並列転送を行う方式はグリッドコンピューティングに限らず、WWWにおけるHTTPでのファイル転送にも活用されており、効率的なファイル転送方式の一種として認識されている[5]。

GridFTPはファイルの送信側・受信側で複数本のTCP接続を結び、ファイル断片（以降チャンクと呼ぶ）を接続毎に並列に転送する。しかし、GridFTPにはTCP接続数の明確な定義は無く、パラメータとして指定する必要がある。また、最適なTCP接続数はネットワークの環境により定まる。TCP接続数を増加させることは、そのままスループット向上には繋がらな

いことが、グリッド標準化団体OGF（Open Grid Forum）により示されている[6]。また、ネットワーク環境は都度変化するものであり、転送中においてもTCP接続数の最適値は常に変動し続ける。GridFTP-APTはこの問題を解決するため、転送中にスループットを計測し、TCP接続数を調整するアルゴリズムを提案している。

GridFTPの特徴として、定常状態においてスループットは、TCP接続数に関して図2のような上に凸の関数となることが、関連研究より明らかになっている[7]。

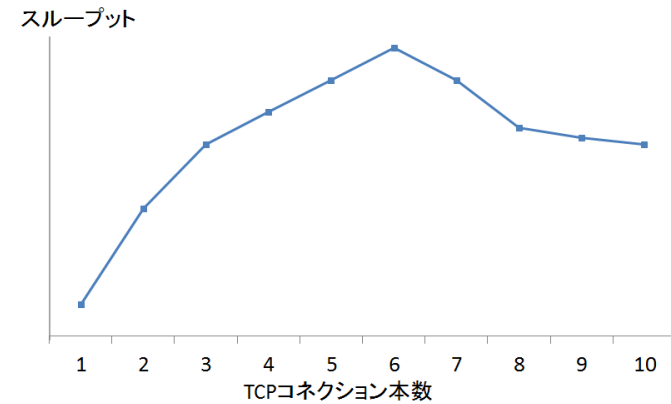


図2 TCP接続数とスループットの関係

GridFTP-APTはこの特性を用い、チャンクを送信しながらスループットの最大値を含む範囲を検索し、その範囲に黄金分割探索法[8]を適応することで、スループットの最大値を特定するアルゴリズムである。

## 3. 研究の目的

本研究では、経路を並列同時に扱うことで転送速度を向上するCMTと、TCP接続数を調整しネットワーク帯域を有効活用するGridFTP-APTを組み合わせることで、ファイル転送処理の高速化を図り、マルチホーム端末同士のアドホック通信における高速なファイル転送方式の実現を目指す。

CMTおよびGridFTP-APTは有線のネットワークを前提に考えられており、本研究の想定環境であるアドホック無線通信は考慮されていない。アドホック無線通信は有線ネットワークにおける通信とは異なる特徴を持ち、特に輻輳制御においてその

差が顕著に表れる[9]. TCPにおいては一般的に、パケットロスや輻輳発生時のシグナルと見なす。パケットロスには輻輳を原因とするものと、衝突などで発生するビット誤りを原因とするものの2種類が存在する。TCPではこれらの原因の区別は行わないため、ビット誤りによるパケットロスが発生した場合、輻輳の誤検出を引き起こしてしまうこととなる。現在の有線ネットワークは回線の信頼性が高く、通信中のビット誤りによるパケットロスが発生する確率は低いため、この誤検出の問題は無視されるが、有線ネットワークに比べ回線の信頼性が低いとされるアドホック無線通信では、輻輳の誤検出が頻発してしまう。TCPは輻輳を検出すると、輻輳制御のアルゴリズムに従い送信量を減らすため、誤検出が起きた場合、送信量が少ないうまになってしまう。これがアドホック無線通信におけるTCPの輻輳制御問題である。

しかし本研究において、この問題はマルチホップでのアドホック無線通信で顕著になる問題であると考えられる。本研究の想定環境である、隣接端末の1hopでの通信の場合、マルチホップの場合に比べビット誤りによるパケットロスの発生率は低く、輻輳の誤検出が多発することはないと考えられるためである。

本研究ではまず、アドホック無線通信環境においてCMTおよびGridFTP-APTが、この輻輳の誤検出の問題などに影響されず、有効に働くことを、予備実験によって検証する。

#### 4. 予備実験および評価

CMTとGridFTP-APTは、有線ネットワークを想定環境としたものである。本章では、本研究の想定環境であるマルチホーム端末同士のアドホック無線通信においてもこれらの技術が有効であることを、実験を行い確認する。

##### 4.1 実験環境

表1表2の端末を用意し、近接させた状態で端末間でのファイル送受信を行った。

表1 実験用端末1 (ファイル送信側)

CPU	Intel Core i5-2450M 2.5GHz
メモリ	4GB
OS	Ubuntu11.04
無線ネットワークデバイス1	Intel PRO/Wireless 2200BG
無線ネットワークデバイス2	BUFFALO WLI-UC-GN

表2 実験用端末2 (ファイル受信側)

CPU	Intel Core Duo U2400 1.06GHz
メモリ	1GB
OS	Ubuntu11.04
無線ネットワークデバイス1	Intel PRO/Wireless 2200BG
無線ネットワークデバイス2	BUFFALO WLI-UC-GN

いずれの端末も無線ネットワークデバイスを2つずつ搭載した無線マルチホーム端末であり、アドホックモードで動作している。また、実験用のプログラムはGCCでコンパイルを行い、並列処理部分にはOpenMP[10]を用いている。

##### 4.2 アドホック無線通信環境におけるCMTの検証と考察

アドホック無線通信環境におけるCMTの有効性を確認するため、従来のパス1本による転送と、パス2本を使用したCMTでの転送について、ファイル転送時のスループットを比較した。また、各パスに対してTCPコネクションは1本ずつとした。転送するファイルの種類はバイナリファイルで、10MBと100MBの2種類を用意した。各10回ずつファイル転送を行い、平均スループットを比較している。結果を図3に示す。

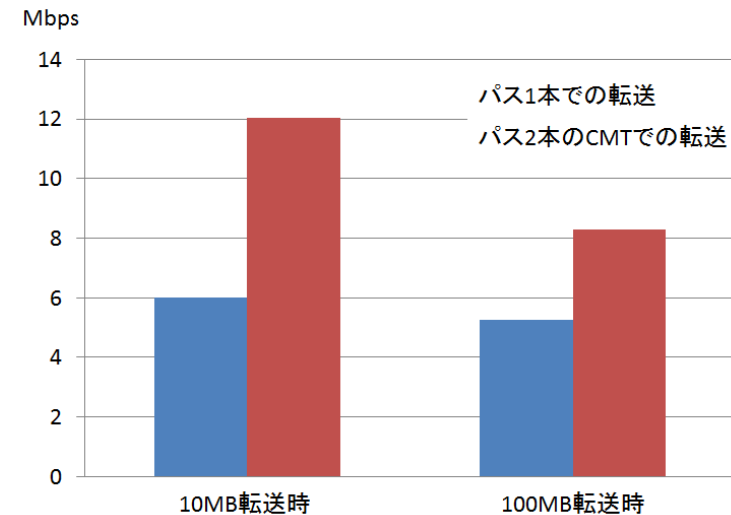


図3 転送方式別スループット比較

実験結果から、CMTによる転送では従来のパス1本による転送方式よりも、1.5倍から2倍近くスループットが向上できていることが分かる。これより、アドホック無線通信環境においても、CMTは有効であると言える。

一方、10MB転送時に比べ、100MB転送時のCMTのスループット向上率は低い。これはファイルサイズが大きくなるにつれ、ファイル断片の欠損確認や順番制御処理の負荷が高くなるためと考えられる。

#### 4.3 アドホック無線通信環境におけるGridFTP-APTの検証と考察

GridFTP-APTは、GridFTPにおける最適TCPコネクション数を求めるアルゴリズムである。アドホック無線通信環境においてGridFTPを行う際に、最適なTCPコネクション数は存在するのか、つまり転送中にコネクション数を変動させる意義はあるのかを、本節では確認する。

実験方法として、GridFTPと同様に、ファイルをチャンクに分割して端末間で転送を行う。チャンクの並列転送を行うTCPコネクション数を1から10まで変動させ、コネクション数ごとのスループットを計測して、その変動を確認する。チャンクサイズは1024Bに固定し、トータルサイズ10MBと100MBの2種類の場合について実験を行った。評価を10回行い、その平均スループットを比較している。

結果を図4 TCPコネクション数ごとのスループット変動に示す。

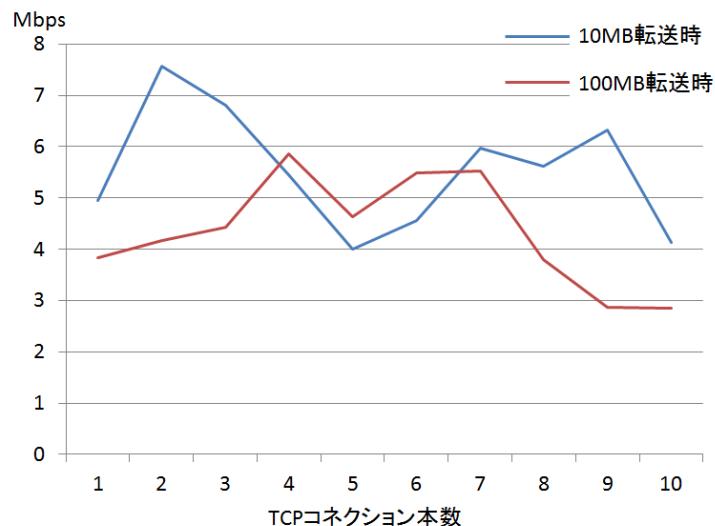


図4 TCPコネクション数ごとのスループット変動

実験結果から、TCPコネクション本数とスループットには相関があることが分かる。今回の実験時における100MB転送時の最適なTCPコネクション本数は4本で、最も低かった10本の際に比べ、2倍近いスループットが得られている。また、スループットの変動が山形を描いていることから、TCPコネクション本数は多すぎても少なすぎても、スループットに悪影響を及ぼすことが分かる。最適なコネクション本数は1本からN本の間に存在しており、最適なコネクション本数を求めることが転送効率の向上に繋がるため、アドホック無線通信環境においてもGridFTP-APTのアルゴリズムは有効であると言える。

また、10MB・100MBどちらの場合においても、スループットの大きな変曲点が3つ存在し、M型の山を描いていることが分かる。GridFTP-APTのアルゴリズムではスループットの変曲点は1つ、つまり凸型の山でなくてはならず、M型の山を探索した場合、コネクション本数の最適値を誤検出してしまう場合が考えられる。

## 5. 提案方式

本章では、提案方式である、GridFTP-APTとCMTを組み合わせた、マルチホーム端末同士のアドホック無線通信における効率的なファイル転送方式を説明していく。

### 5.1 制約条件

本提案方式は、アドホック環境への適応およびアルゴリズムの簡略化のため、以下の項目を制約条件とする。

- 無線ネットワークデバイスは、送信側・受信側ともに $r$ 個であり、互いに既知とする。本提案は隣接端末で行うため、無線ネットワークデバイスの個数は利用ユーザが確認できるためである。
- チャンクサイズは固定とする。
- 転送に使用するコネクション本数の上限 $P$ を設ける。転送開始時に管理用コネクションと合わせて $P+1$ 本のTCPコネクションを確立し、転送終了まで解放しない。そのため、転送中に使用しないコネクションが生まれる。

### 5.2 概要

GridFTP-APTはチャンクを送信しつつコネクション数を調整・変動させるアルゴリズムであり、CMTは2つ以上のネットワークデバイスで同時に送受信する技術である。これらを組み合わせた処理の流れを図5に示す。チャンク分割・結合操作やスケジューリングなど、ファイル転送全体の管理をCMTコントローラが行い、送信の処理及びコネクション本数の調整処理をapt-sendが、受信処理をapt-recvがそれぞれ行う。送信側のCMTコントローラは、無線ネットワークデバイスの数だけapt-sendのインスタンスを持ち、受信側も同様にapt-recvのインスタンスを持つ。

apt-send と apt-recv 間では  $k$  本の TCP コネクションを同時にチャンク転送に用いる。 $k$  の値は GridFTP-APT のアルゴリズムに従い変動する。

また、apt-send と apt-recv 間には管理用コネクションが 1 本存在する。これはファイル転送の制御用メッセージ送受信に用い、チャンク送受信には用いない。

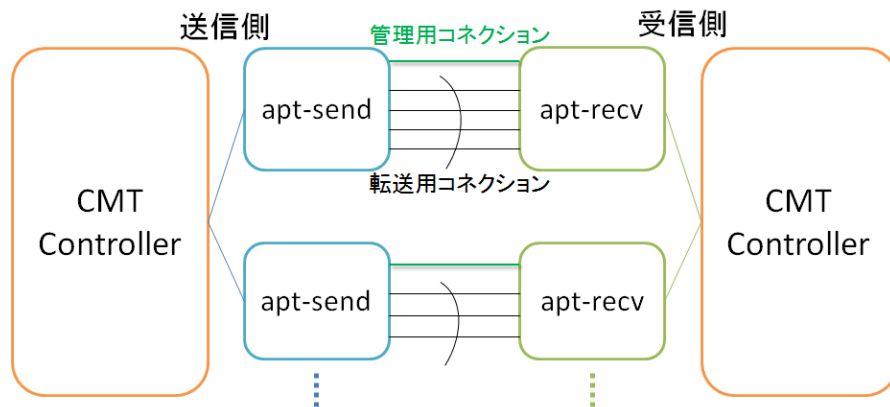


図 5 送受信側関係図

### 5.3 転送開始処理

転送開始の処理は CMT コントローラが行う。送信側 CMT コントローラはまず、転送対象のファイルをチャンクに分割する。各チャンクには番号づけを行い、番号順に結合することで元のファイルに復元できるようにする。また、チャンクのサイズは、各チャンクの番号を含めてパケットと同程度とする。これはスループットの変動が起こった際に、使用するコネクション数を即座に変動可能とするためである。

次に、送受信側それぞれで apt-send と apt-recv のインスタンスを  $r$  個生成し、それぞれのインスタンスで  $P+1$  本分のコネクション確立処理を行う。コネクション確立処理完了後、送信側 CMT コントローラは 1 つ目の apt-send の管理用コネクションから、チャンクの総数を受信側に送信する。受信側 CMT コントローラはチャンクの総数を受け取った後、確認メッセージを同じ apt-recv の管理用コネクションから返答する。送信側 CMT コントローラが確認メッセージを受け取り、転送フェーズに移行する。

### 5.4 転送中の処理

チャンクの到着状況の管理など、転送についての総合的な管理は送信側 CMT コントローラが行い、apt-send では GridFTP-APT アルゴリズムを適用して実際の送信を行う。送信側 CMT コントローラは未送信のチャンクを複数個 apt-send に渡し、

apt-send は渡されたチャンクを各コネクションに対して、それぞれ同時並列に送信する。CMT コントローラが何個のチャンクを apt-send に渡すかは、apt-send が GridFTP-APT のアルゴリズムで求めた、使用するコネクション数  $k$  の値に従う。apt-send は  $k$  個のチャンクを送信後、スループットを求め、GridFTP-APT のアルゴリズムに従い  $k$  の値を再計算し、送信側 CMT コントローラに報告する。送信側 CMT コントローラは、apt-send に再計算した  $k$  の個数分だけチャンクを引き渡す。この操作を apt-send インスタンス数分だけ同時並列に行う。これを、チャンクを全て送り終わるまで続けていく。なお、 $k$  の初期値は GridFTP-APT のアルゴリズムに従い 1 とする。

チャンクの受信は apt-recv が行い、apt-send から受け取ったチャンクを受信側 CMT コントローラに引き渡す。受信側 CMT コントローラはチャンクに付けられた番号順にチャンクを格納する。数回の受信後にチャンクの欠番があった場合、いずれかの apt-recv の管理用コネクションより再送要求を送信する。再送要求には欠けているチャンクの番号すべてが記載される。再送要求を受け取った送信側 CMT コントローラは、再送要求に記載されている番号のチャンクを apt-send に引き渡す。

送信側 CMT コントローラがすべてのチャンクを送信後、転送終了フェーズに移行する。

#### 5.4.1 アドホック無線通信環境への適応

アドホック無線通信環境においては、コネクション数に対してのスループットの変動が激しく、M 型になりやすいことが 4.3 節で明らかになった。従来の GridFTP-APT は凸型になることを前提に設計されているため、頂点が 2 つの場合への対応が必要となる。

GridFTP-APT は、最大値の探索に黄金分割探索法を用いる。黄金分割探索法は、最大値を含んだ上に凸になる範囲（以降ブラケットと呼ぶ）に適応させる探索法である。そのため、GridFTP-APT はまず、あるコネクション数のときのスループットの値を求め、その変動からブラケットを求める。次に、求めたブラケットに黄金分割探索法を適応し、スループットが最大となる最適な TCP コネクション数  $k$  を求めていく。GridFTP-APT のブラケット探索手順を、図 6 を用いて説明する。

GridFTP-APT はコネクション数  $(x, y, z)$  を増加させつつ転送を行い、コネクション数  $z$  のときのスループットが、コネクション数  $y$  のときを下回ったときに、 $(x, y, z)$  をブラケットと確定する。しかし、図 7 のように  $z$  よりも多い TCP コネクション数でスループットが最大となった場合、最大値が発見できない。



スループット

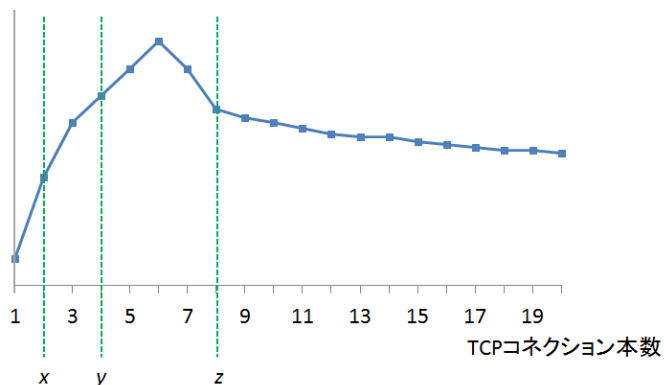


図 6 GridFTP-APT のブラケット探索手順

スループット

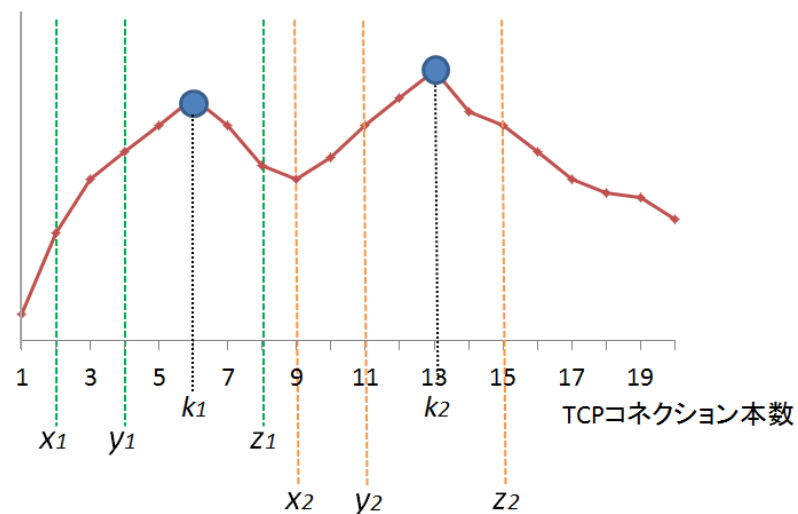


図 8 提案方式のブラケット探索手順

スループット

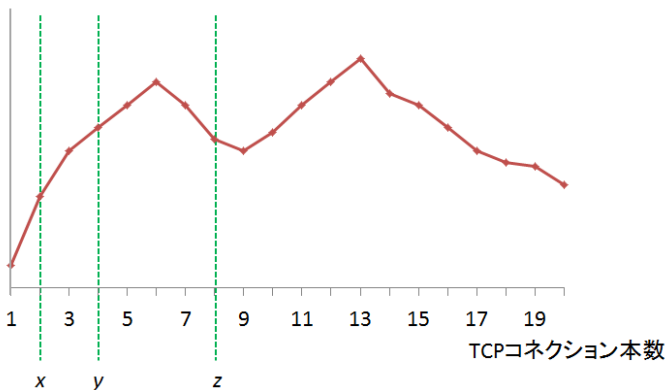


図 7 最適なブラケット探索失敗の様子

提案方式では、ブラケットの探索を複数回行うことで、スループットの変動が M 型の山を描く場合に対応している。1 つ目のブラケット  $(x_1, y_1, z_1)$  が確定したあと、コネクション数の初期値を  $x_2 = z_1 + 1$  として、2 つ目のブラケット  $(x_2, y_2, z_2)$  の探索を開始する。ブラケット  $(x_2, y_2, z_2)$  探索の略図を図 8 に示す。

2 つ目のブラケット  $(x_2, y_2, z_2)$  探索後、同様の手順で 3 つ目以降のブラケットを  $n$  個目まで探索していく。全てのブラケットを探索し終えた後、従来の GridFTP-APT と同様に黄金分割探索法を用い、全てのブラケットに対してコネクション数の最適値  $k_1 \sim k_n$  を探索する。最終的に、 $k_1 \sim k_n$  の値を比較して、スループットが最も大きいものを最適なコネクション数  $k$  として確定する。

2 つ目以降のブラケット探索時に  $z_n > P$  となった場合、そこでブラケット探索を中止し、これまでのブラケットから最適なコネクション数  $k$  を探索して、最適値として確定する。この手順では従来の GridFTP-APT に比べ、スループットの最大値  $k$  を得るまでの探索回数が増えることとなるが、ファイルサイズが大きいほど最大スループットで転送できる割合が増えるため、想定する数十 MB～数 GB のファイル転送においては有効となり得る。

#### 5.4.2 障害発生時の対応

送信側 CMT コントローラは apt-send に対し、強制的にコネクション本数を変更させる権限を持つ。これは転送中に急激なスループットの変動が起きたとき、つまり転送中の障害に対応するためである。CMT コントローラは apt-send から報告を受け

## 参考文献

たコネクション本数とスループットを記録しておく。今回のコネクション本数  $k_n$  における送信時のスループットが、前回に比べ急激に減衰した時、CMT コントローラは前回のコネクション本数  $k_{n-1}$  を apt-send に渡す。この時 apt-send は、コネクション本数の初期値を  $k_{n-1}$  に設定し、GridFTP-APT のアルゴリズムを再開する。再開後、apt-send から報告されたスループットが減衰前より改善されない場合、 $k_{n-1}$  をデクリメントして再度 apt-send に引き渡す。 $k_n$  が 0 になった場合、その apt-send からはチャンクは送信されないこととなる。

### 5.5 転送終了処理

送信側 CMT コントローラは、いずれかの apt-send の管理用コネクションから転送終了メッセージを送信する。転送終了メッセージを受け取った受信側 CMT コントローラは、チャンクの総数が開始時に伝えられた値と等しいこと、欠番がないことを確認し、完了メッセージを送信する。欠番があった場合、再送要求を送信する。全てのチャンクが揃ったことを確認後、受信側 CMT コントローラはチャンクの結合処理を行う。

送信側 CMT コントローラが完了メッセージを受信後、全ての apt-send インスタンスでコネクション解放処理を行い、送信完了となる。

## 6. おわりに

本稿では、マルチホーム端末同士のアドホック無線通信における効率的なファイル転送方式を提案した。CMT と GridFTP-APT はいずれも有線ネットワークを前提に効率的なファイル転送を実現するよう考案されたものであり、アドホック無線通信環境においての有効性は不明確であった。そのため、これらの技術について、アドホック無線通信環境での実験を行い、その有効性を確認した。

また実験から、コネクション数に対するスループットの変動は、アドホック無線通信においては M 字型になりやすいことが明らかになった。提案方式ではこの特性に対処するため、GridFTP-APT のアルゴリズムを改良し、ブラケット探索を  $n$  回行うという手法を取り入れた。

これによって、従来の GridFTP-APT では探索不可能なコネクション最適値を求めることが可能となる。一方、コネクション最適値を求めるまでの探索回数が増加するという問題も発生し、最適でないコネクション数での転送機会が増え、ファイル全体の転送に要する時間が増大する可能性がある。

今後は実験などによって、様々な場合におけるスループットの増減傾向を明らかにし、提案手法の改善をはかる。また、転送効率の向上について従来のファイル転送方式との比較を行う。

- [1] Janardhan R. Iyengar, Paul D. Amer, Randall Stewart, "Concurrent Multipath Transfer Using Transport Layer Multihoming: Performance Under Varying Bandwidth Proportions," IEEE Military Communications Conference Vol.1, pp.238-244, 2004
- [2] 伊藤 建志, 大崎 博之, 今瀬 眞 "GridFTP-APT : データ転送プロトコル GridFTP の並列 TCP コネクション数調整機構", 電子情報通信学会技術研究報告. IN, 105 (472) , pp.19-24, 2005
- [3] "GridFTP v2 Protocol Description," <http://www.ogf.org/documents/GFD.47.pdf>
- [4] 柏原 貴之, 中里 秀則, "GridFTP における並列転送方式の高速化を実現する輻輳制御方式に関する研究", 電子情報通信学会技術研究報告. IN, 情報ネットワーク 109 (411) , pp.87-92, 2010
- [5] 佐竹 伸介, 稲井 寛, 荒井 剛, "Web サーバシステムにおけるコネクション受付制御方式", 電子情報通信学会技術研究報告. NS, 107 (403) , pp.37-42, 2007
- [6] "The Globus Alliance," <http://www.globus.org/>
- [7] 伊藤 建志, 大崎 博之, 今瀬 眞, "広域グリッドコンピューティングにおけるデータ転送プロトコル GridFTP のパラメータ設定方法に関する検討", 電子情報通信学会技術研究報告. IN, 情報ネットワーク 104 (182) , pp.19-24, 2004
- [8] PRESS W. H, Numerical Recipes in C : The Art of Scientific Computing, 1992
- [9] Oyunchimeg Shagdar, Mehdad N. Shirazi, 湯 素華, 鈴木 龍太郎, 小花 貞夫, "無線アドホックネットワークにおける分散型輻輳制御", 電子情報通信学会技術研究報告. RCS, 無線通信システム 108 (117) , pp.1-5, 2008
- [10] "The OpenMP API," <http://openmp.org/wp/>