Vol.2013-MBL-68 No.16
Vol.2013-ITS-55 No.16
Vol.2013-DCC-5 No.16
2013/11/15

アドホック通信における高速ファイル転送方式の提案

高橋大斗[†1]　中村嘉隆[†2]　高橋修[†2]

近年のモバイル端末の高性能化に伴い，モバイル端末が扱うファイルサイズは肥大化している．ファイルサイズの肥大化への対応のため，アドホック通信を用いたモバイル端末間のファイル転送方式は今後さらなる改良が求められる．また，モバイル端末間でのファイル転送の際には，中継役を不要とするアドホック通信が適していると考えられる．本研究ではアドホック通信におけるファイル転送効率化の手段として，マルチホーミングと TCP マルチコネクションに焦点を当て，モバイル端末間における効率的なファイル転送方式を提案・実装した．本提案方式の有効性を示すべく，無線通信環境の変化時におけるスループットの変動などにも注目し，その評価を行った．

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課題となる．

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が考えられるが，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．一方，端末同士が直接行うアドホック無線通信では，ファイルのサイズ上限などの問題は発生しない上に，近年の無線ネットワークデバイスの普及・低価格化に伴って一台のモバイル端末に複数の無線ネットワークデバイスを持つマルチホーム端末も一般化してきていることから，アドホック無線通信を用いたファイル転送は隣接モバイル端末間でのファイル交換に対して有効であると考えられる．

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．本研究ではファイル転送高速化の手法として，Concurrent Multipath Transfer (CMT) [1] と GridFTP-APT [2] に注目，これらを組み合わせた，アドホック無線通信に適した高速なファイル転送方式の提案および実装を行って，その有効性の評価を行ってきた[3].

本稿では提案ファイル転送方式に対し，ネットワーク障害の発生時など，転送中の著しいスループット変動時における処理の追加を行い，その評価を行った．

†1 公立はこだて未来大学大学院 システム情報科学研究科
Graduate School of Systems Information Science, Future University Hakodate
†2 公立はこだて未来大学 システム情報科学部
School of Systems Information Science, Future University Hakodate

2. 関連技術

本章では，本研究で注目した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].

送信プロセスがパケットを送信する際，パス 1 本のみを用いる従来の転送方式では，パスへの引き渡しやエラーチェックなどのネットワークデバイスの準備が整うまで，送信プロセスはネットワークデバイスへ次のパケットを引き渡すことができない．一方，CMT による転送方式ではパスを 2 本以上同時並列に用いることで，送信プロセスが一度にネットワークデバイスへと引き渡せるパケットが増えるため，転送量増加に繋がる．シングルパスによる従来の転送方式と，パス 2 本を用いた CMT による転送方式の比較を図 1 に示す．

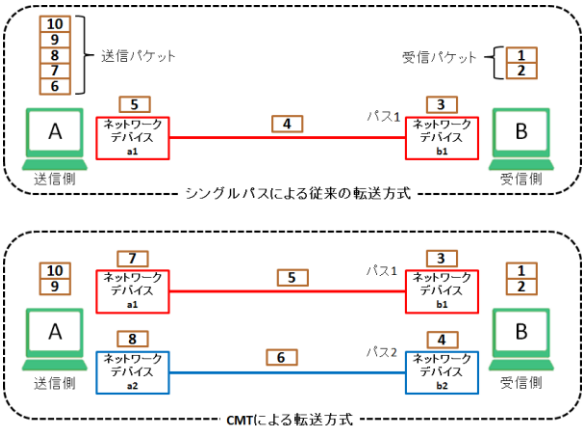

図 1 転送方式の比較

Vol.2013-MBL-68 No.16
Vol.2013-ITS-55 No.16
Vol.2013-DCC-5 No.16
2013/11/15

2.2 GridFTP-APT

GridFTP-APT [2]とは, GridFTP [4]における TCP コネクション数の最適値を求め，TCP マルチコネクションの利点を最大限活用す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 コネクション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．なお，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ったものではなく，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 コネクションを結び，ファイル断片（以降チャンクと呼ぶ）をコネクション毎に並列に転送する．しかし，GridFTP には TCP コネクション数の明確な定義は無いため, パラメータとして指定する必要がある．また，最適な TCP コネクション数はネットワークの環境により定まる．TCP コネクション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]．さらに，ネットワーク環境は常に変化するものであり, 転送中においても TCP コネクション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 コネクション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，転送時のスループットは TCP コネクション数に応じて変化し，スループットを最大化する TCP コネクション数が存在することが関連研究より明らかになっている[7]. GridFTP における TCP コネクション数とスループットの関係の例を図 2 に示す[2].

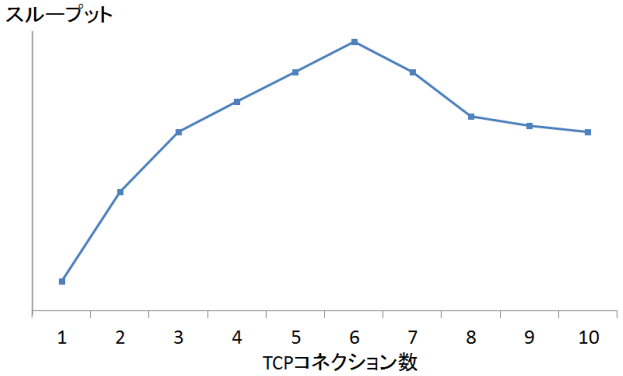

図 2 TCP コネクション数とスループットの関係（例）

GridFTP-APT は図 2 の特性を利用し，チャンクを送信しながらスループットの最大値を含む TCP コネクション数の範囲を探索して，その範囲に黄金分割探索法[8]を適応することで, スループットが最大となる TCP コネクション数を特定するアルゴリズムである．

3. 提案方式

本章では，本研究で提案している 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の概要を説明する．

3.1 制約条件

本提案方式では，アドホック環境への適応とアルゴリズム簡略化のため，以下の項目を制約条件とする．

(1) 無線ネットワークデバイスの個数

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同士が確認できる．

(2) チャンクサイズ

チャンクサイズは固定とし, 転送対象のファイルの 1/100 とする．

(3) コネクション上限の設定

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．

3.2 送信側と受信側の接続

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．送受信時におけるこれらの処理の概要を図 3 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 Controller 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 Controller 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 間には転送用コネクションが k 本存在し，これら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のみ用い，チャンク送受信には用いない．

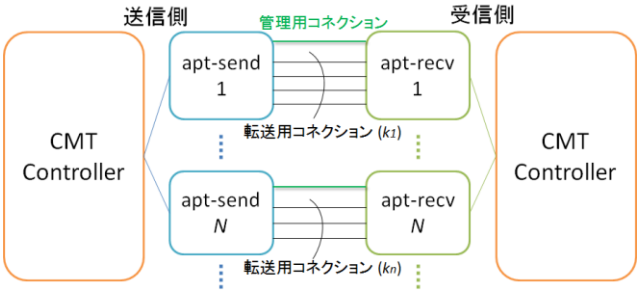

図 3 送信側と受信側の関係図

Vol.2013-MBL-68 No.16
Vol.2013-ITS-55 No.16
Vol.2013-DCC-5 No.16
2013/11/15

3.3 転送開始処理

転送開始の処理は CMT Controller が行う．送信側 CMT Controller はまず，転送対象のファイルをチャンクに分割する．各チャンクに順序番号を付し，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．

次に，送受信側それぞれで apt-send と apt-recv のインスタンスを各々 R 個，つまりネットワークデバイスの数だけ生成し，それぞれのインスタンスで $P+1$ 本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 Controller は 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 Controller 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 Controller が確認メッセージを受け取り，転送処理に移行する．

3.4 転送処理

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 Controller 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 Controller 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，それぞれ並列同時に送信する．CMT Controller 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 k の値に従う．apt-send は k 個のチャンクを送信後に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 Controller に報告する．送信側 CMT Controller は，再計算した k の個数分だけ apt-send にチャンクを引き渡す．この操作を R の数だけ同時並列に行い，チャンクを全て送り終えるまで続ける．なお，k の初期値は GridFTP-APT のアルゴリズムに従い 1 とし，上限は $P+1$ となる．

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 Controller に引き渡す．受信側 CMT Controller 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 Controller 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 Controller 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．

3.5 転送終了処理

送信側 CMT Controller が 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し，転送終了処理開始となる．転送終了メッセージを受け取った受信側 CMT Controller 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 Controller はチャンクの結合処理を行う．

送信側 CMT Controller 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．

3.6 転送中の例外処理

アドホック無線通信は，電波干渉など周囲からの影響により，転送中のネットワーク障害が発生しやすい環境にある．ネットワーク障害が発生した場合，スループットの著しい低下が現象として現れる．本提案方式ではネットワーク障害が発生した場合にも，マルチパスの利点を活かし最適な転送を行えるような処理を組み込んでいる．以降，この処理を例外処理と呼ぶ．

例外処理は apt-send と送信側の CMT Controller が行い，GridFTP-APT アルゴリズムにおける k の値の算出・報告のタイミングで次の処理を決定する．apt-send および送信側 CMT Controller の例外処理判定のフローを，図 4 と図 5 に示す．

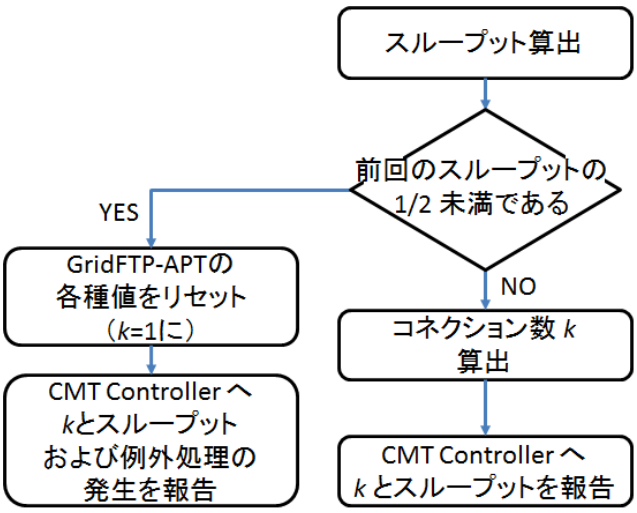

図 4 例外処理：apt-send のふるまい

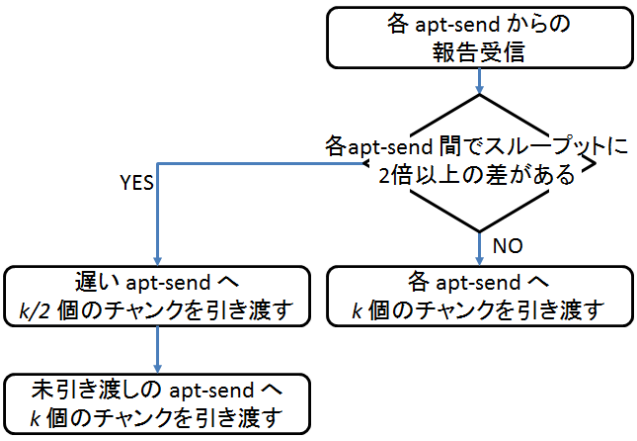

図 5 例外処理：CMT Controller のふるまい

Vol.2013-MBL-68 No.16
Vol.2013-ITS-55 No.16
Vol.2013-DCC-5 No.16
2013/11/15

GridFTP-APT のアルゴリズムでは，チャンクの送信毎にスループットを算出し，利用するコネクション数 k を求める．本提案方式ではこれを一つの区切りと考える．

apt-send はこの区切りのたびに前回のスループットとの比較を行い，著しい低下（本提案方式では前回と比較して 1/2 を下回ったときとする）が認められた場合，ネットワーク環境の変化が起こったとみなし，GridFTP-APT で求めるコネクション数の管理に用いる値をすべてリセットし，コネクション数 k を再度計算し直す．

CMT Controller はこの区切りのたびに各 apt-send, つまりネットワークデバイスごとのスループットを比較し，スループットに大きな差異（本提案方式では最も高いスループットを持つネットワークデバイスと比較して 1/2 を下回ったときとする）が認められた場合，ネットワークデバイスの障害発生と認識し，遅いデバイスには通常割り当てるべきチャンクの数を，半分にすることで対応する．

これらの処理を複合して行い，早く正常に動くネットワークデバイスにはチャンクを多く，遅く異常が認められるデバイスにはチャンクを少なく，意図的に割り当てることにより，マルチパス・マルチコネクションを有効活用した柔軟な転送方式を実現している．

4. 提案方式の評価

本章では，提案方式の評価と，その考察を行う．

評価実験として，ファイル転送にかかった転送時間を計測し，標準的なファイル転送プロトコルである FTP との比較を行った．本研究は大容量のファイル交換をターゲットとしているため, 転送するファイルサイズは 1GB と設定した．また，転送中に片方のネットワークに障害が発生するという場合を想定した実験を行い，例外処理の有効性を確かめた．

4.1 実験環境

表 1 に挙げたスペックの 2 つの端末間でファイル転送することで，評価実験を行った．同種のネットワークデバイスを 2 つ持つ，マルチホーム端末である．

通常の転送時に周囲からの電波干渉などが発生しない環境を実現するため，端末は仮想マシンで作成し，仮想ネットワーク上で実験を行っている．各ネットワークデバイスは無線 LAN 規格の一つである IEEE802.11g を想定するため，転送速度の上限を設け，11Mbps を上限としている．また，仮想マシンおよび仮想ネットワークの作成には VMware Player [9]を用いている．

表 1 仮想マシンのスペック

CPU コア数	2
メモリ	1GB
OS	Ubuntu 11.04
ネットワークデバイス 1 の速度上限	11Mbps
ネットワークデバイス 2 の速度上限	

4.2 FTP との比較実験

本節ではマルチパス・シングルコネクション（CMT）と，提案方式であるマルチパス・マルチコネクション（CMT + GridFTP-APT）を用いたファイル転送の有効性を示すため，標準的なシングルパス・シングルコネクションのファイル転送プロトコルである FTP との比較実験を行った．各実験環境における設定を図 6，図 7，図 8 にそれぞれ示す．

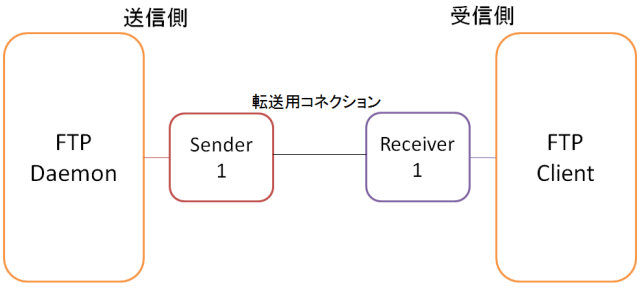

図 6 FTP 実験時の設定

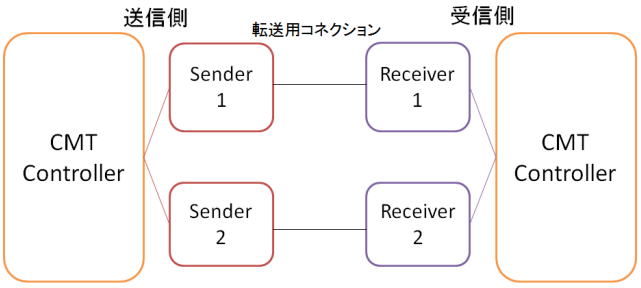

図 7 CMT 実験時の設定

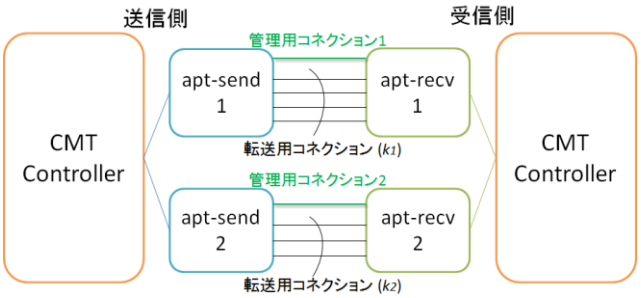

図 8 提案方式実験時の設定

Vol.2013-MBL-68 No.16
Vol.2013-ITS-55 No.16
Vol.2013-DCC-5 No.16
2013/11/15

1GB のファイルを FTP で転送した際にかかった時間を 100%として，各方式との割合を表 2 に示す．なお，FTP デーモンには vsftpd [10]を用いている．

表 2 FTP との転送時間比較結果
（ネットワーク遅延・パケットロス等無しの場合）

方式	FTP を 100%とした時の割合
CMT	199.1 %
提案方式	199.9 %

4.3 ネットワーク障害発生時の比較実験

本節では例外処理の有効性を示すため，ネットワーク障害発生時と，正常動作時（4.2 節で得られた結果）との比較実験を行った．

実験のシナリオとして，正常時の転送にかかる時間のおよそ半分である 180sec 後に，ネットワーク 2 に障害を発生させる．ネットワークは完全には遮断されず，送受信は可能ではあるが，ネットワークデバイス 2 のスループットが約 1/25 まで低減するという条件で実験を行う．これにより，スループット低下を検知した際の apt-send のふるまい，ネットワークデバイス間の速度差を検知した際の CMT Controller のふるまいを検証する．

実験にはネットワークエミュレータの一種である netem [11]を用いた．転送開始から 180sec 後に，netem を用いネットワークデバイス 2 へ 3000ms のディレイをかけ続けることで，本シナリオを実現している．

正常時と本シナリオ時で，まず CMT についての比較を行った．次に，例外処理を用いない際の提案方式の比較，最後に例外処理を用いた際の比較を行った．表 3 に，各方式の正常時の転送時間を 100%とした割合を示す．

表 3 ネットワーク障害発生時の転送時間比較結果

方式	正常時を 100%とした時の割合
CMT	13.7 %
提案方式 例外処理無効の場合	57.0 %
提案方式 例外処理有効の場合	66.6 %

例外処理が発生することで，転送中のコネクション数が強制的に書き換えられ，最終的に各 apt-send に割り当てるチャンクの総数が変わることになる．3.4 節で述べたとおり，CMT Controller が apt-send に引き渡すチャンクの個数は，apt-send が現在利用する TCP コネクションの本数と等しくなるためである．図 9 と図 10 に，例外処理無効時と有効時それぞれのコネクション数の推移を，表 4 に最終的に各 apt-send が転送したチャンクの総数を示す．

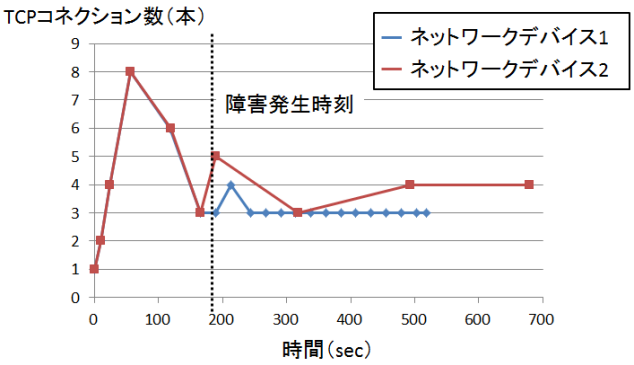

図 9 コネクション数推移：例外処理無効の場合

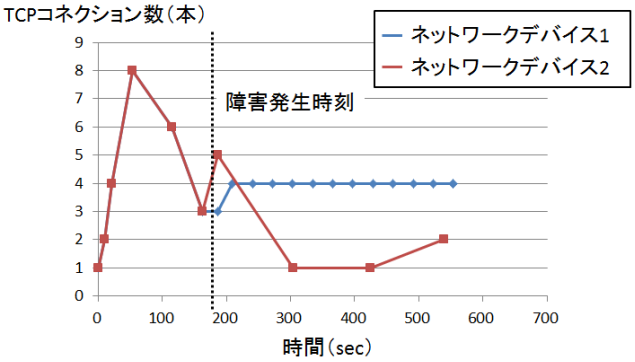

図 10 コネクション数推移：例外処理有効の場合

表 4 各デバイスで送信したチャンクの個数

方式	ネットワークデバイス 1	ネットワークデバイス 2
CMT	50	50
提案方式 例外処理無効の場合	62	38
提案方式 例外処理有効の場合	71	29

図 10 から，例外処理有効の場合では，ディレイがかかった直後にネットワークデバイス 2 のコネクション数が初期化され，1 本に戻っていることが見てとれる．また，表 4 から，例外処理有効の場合にはネットワークデバイス 2 へのチャンクの割り当てが著しく減っていることが解る．

4.4 評価・考察

CMT，および提案方式の結果は，FTP と比較していずれもおよそ 2 倍の転送速度向上が認められた．シングルパスの FTP と，2 本のマルチパスでの CMT および提案方式との比較であるため，妥当な結果と言える．

一方 CMT と提案方式との比較では，FTP を基準としてコンマ数%程度の向上しか認められないが，ネットワーク障害発生時の低減率を鑑みることで提案方式の優位性が確認できる．表 3 の正常時を基準とした割合において，CMT

と提案方式の例外処理有効の場合とを比較すると，提案方式が 50%以上，正常時の転送時間に近いことがわかる．このように通信障害発生時にも，提案方式は有効に働いている．

また，表 3 の結果より，例外処理無効時に比べ，有効時では 10%近くの向上が見られる．例外処理有効の場合，異常が認められるネットワークデバイスの利用率が大幅に下がるため, 正常なネットワークデバイスの利用率が上がり，結果として全体の転送効率が向上したためと考えられる．この結果は，表 4 の各方式におけるチャンク送信個数からもわかる．例外処理無効の場合と有効の場合では，送信個数に 10 個近い差が見られる. チャンクのサイズは送信対象のファイルの 1/10 としたため，例外処理有効の場合には，100MB 近くが正常なネットワークデバイスに多く割り当てられている．早く正常に動くネットワークデバイスにはチャンクを多く，遅く異常が認められるデバイスにはチャンクを少なく，意図的に割り当てる，という例外処理の目的を果たし，その有効性を証明したと言える．

5. まとめ

本研究では，隣接し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高速なファイル転送方式を提案した．提案方式では CMT と GridFTP-APT という，想定環境に対して親和性の高い 2 つの技術に着目し，これらの利点を活かせるアルゴリズムを実現した．さらに，転送中のネットワーク障害が起こりやすいアドホック無線通信環境に適応するための例外処理を提案方式に追加した．

従来のファイル転送方式である FTP との比較実験では，提案方式に 2 倍近くの向上が見られた．また，転送中に片方のネットワークに障害が発生した場合を想定し，ネットワークエミュレータを用いた実験を行うことで，例外処理の有効性を確認した．例外処理を用いた提案方式では，転送効率は CMT に比べ 50%以上，例外処理を用いない場合の提案方式に比べ 10%近く向上しており，例外処理の有効性を示すことができた. また, 提案方式は CMT に比べて，耐障害性の面でも優れた転送方式であることがわかった．

今後は，例外処理の追加改良を含めた提案方式の更なるブラッシュアップを行い，IEEE802.11g と Bluetooth の組み合わせなど，異なる規格のネットワークデバイスを同時に用いた場合の，各規格に適したチャンク割り当て方式などを考慮する必要がある．

参考文献

1) Janardhan R. Iyengar, Paul D. Amer, Randall Stewart, "Concurrent Multipath Transfer Using Transport Layler Multihoming: Performance Under Varying Bandwidth Proportions," IEEE Military Communications Conference Vol.1, pp.238-244, November 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, December 2005.
3) 高橋 大斗, 中村 嘉隆, 高橋 修, “アドホック通信における高速ファイル転送方式の提案と実装, ”電子情報通信学会研究報告. NS, 112（392）, pp.43-48, January 2013.
4) "GridFTP v2 Protocol Description,"
http://www.ogf.org/documents/GFD.47.pdf
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, February 2007.
6) "The Globus Alliance," http://www.globus.org/
7) 伊藤 建志, 大崎 博之, 今瀬 真, “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, July 2004.
8) PRESS W. H, Numerical Recipes in C : The Art of Scientific Computing, Cambridge University Press;, 1992
9) VMware http://www.vmware.com/
10) vsftpd, "Probably the most secure and fastest FTP server for UNIX-like systems," http://vsftpd.beasts.org/
11) netem http://www.linuxfoundation.org/