情報処理学会論文誌 Vol.57 No.2 620–628 (Feb. 2016)

マルチホーム端末におけるアドホック無線通信を用いた高速ファイル転送方式の提案

中村 嘉隆[1,a)] 高橋 大斗[2] 高橋 修[1,b)]

受付日 2015年5月14日, 採録日 2015年11月5日

概要：近年，スマートフォンやタブレットに代表されるモバイル端末の普及が著しい．またモバイル端末の性能向上にともない，端末で取り扱うファイルサイズも肥大化することが考えられる．現在のモバイル端末には無線 LAN などの通信デバイスが搭載されており，今後も需要が増加すると思われる．一方，通信の転送速度に関わる通信路の帯域幅は規格により理論上の上限が定められており，通信の際には規格値以下の値しか得られない．これに対し通信経路やコネクションを並列に用いることで，理論値に近い，あるいは超える通信速度を実現することが可能となる．本論文ではマルチコネクション時の最適な TCP コネクション数を与える GridFTP with Automatic Parallelism Tuning（GridFTP-APT）と end-to-end のマルチパス接続を保証する Concurrent Multipath Transfer（CMT）を組み合わせることで，高速なファイル転送方式を提案する．また，GridFTP-APT アルゴリズムを改良し，通信中の環境変化にも柔軟に対処できる例外処理アルゴリズムを提案方式に取り入れる．提案方式を実装し，従来の転送方式とのスループット比較，および通信中に環境変化が発生した場合のスループットの変化を評価することで有効性を示した．

キーワード：アドホック無線通信，高速ファイル転送，マルチホーム端末，マルチパス，マルチコネクション

A Proposal of the High-speed File Transfer Method Using Ad-hoc Wireless Communication between Multi-home Terminals

Yoshitaka Nakamura[1,a)] Hiroto Takahashi[2] Osamu Takahashi[1,b)]

Received: May 14, 2015, Accepted: November 5, 2015

Abstract: In recent years, the spread of mobile devices such as tablet PCs or smartphones is remarkable. And with the performance improvement of mobile terminals, the file size that terminals deal with is enlarged. Communication devices such as the wireless LAN are equipped current mobile terminals, and the demand for the devices will increase in future. On the other hand, as for the bandwidth of the channel affecting transfer speed of the communication, the theoretical upper limit is established by a standard, and only speed that is less than standard value is provided in the case of the communication. For this problem, we can realize transmission rate that is almost or more than the theoretical value by using communication paths or connections in parallel. In this paper, we propose a high-speed file transfer method by combining GridFTP with Automatic Parallelism Tuning (GridFTP-APT) which gives the most suitable number of TCP connections in the multi-connection and Concurrent Multipath Transfer (CMT) which guarantees multi-path connection of end-to-end together. In addition, we improve GridFTP-APT algorithm and adopt the exception handling algorithm that can flexibly deal with environmental change of communication for the proposed method. We implement the proposed method and show the effectiveness of the method by evaluating throughput variations when environmental changes occurred during communication, and by comparing throughput with the conventional transfer method.

Keywords: ad-hoc wireless communication, high-speed file transfer, multi-home terminal, multi-path, multi-connection

1. はじめに

モバイル機器の普及は近年著しく，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も増加している．このようなファイル交換の際に着目される点は手軽さであり，制約の少なさや，いかに短時間で交換が行えるかが課題となる．

隣接したモバイル機器間でのファイル交換手法としては，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な方法が存在するが，転送可能なファイルサイズに上限が設けられている，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する．端末どうしで直接行うアドホック無線通信では，ファイルサイズの上限などの問題は発生せず，また近年のほとんどのモバイル端末には，無線ネットワークインタフェースが搭載されているため，最も制約の少ないファイル交換手法の1つであるといえる．

また，近年のモバイル機器の高性能化にともなって，モバイル機器が扱うファイルのサイズも肥大化している．従来の無線通信を用いたファイル転送方式では数十 Mbyte から数 Gbyte のファイルを扱う場合には所要時間が著しく増大し，ファイル交換の利便性を損ねるため，ファイル転送の高速化は重視すべき点である．一方，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，1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，いわゆるマルチホーム端末も一般化しつつあり，複数のインタフェースを効率良く利用することで，さらに高速なファイル交換も可能となる．高速化手法としてはパス並列化手法とコネクション並列化手法があげられる．パス並列化とは通信経路を複数用い，同時に送受信を行う帯域幅拡張のための手法である．代表的な例として，ネットワークインタフェースカード（NIC）を束ねて冗長性・帯域幅拡張を行うリンクアグリゲーション [1]，電波を発するアンテナを複数用いる MIMO（Multiple Input Multiple Output），周波数帯を分割したチャネルを複数用いるチャネルボンディングといった技術があげられる．一方，コネクション並列化とは，end-to-end 間で複数の TCP コネクションを確立し，並列にデータ転送を行う手法である．例として，1つの宛先に対し複数の TCP コネクションを確立し，総合的に時間あたりのパケット送信量を増やすことで，TCP のウィンドウ制御アルゴリズムにおけるスロースタートの回避と，帯域の上限への速い到達を目的とした Parallelizing TCP Connections [2] がある．この手法は Grid コンピューティング環境における GridFTP [3] で実装され，実際に用いられている．

そこで本論文では，これらの点に着目し，ファイル転送高速化の手法として，マルチホーム端末どうしが結ぶ複数の経路を同時に用いて帯域幅の拡張を実現する Concurrent Multipath Transfer（CMT）[4]，および，コネクション数の調整を行って帯域を有効活用する GridFTP with Automatic Parallelism Tuning（GridFTP-APT）[5] の親和性の高い2つの技術を利用することで，マルチホーム端末どうしのアドホック無線通信による高速なファイル転送方式を提案する．

[1] 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, Hakodate, Hokkaido 041–8655, Japan
[2] 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, Hakodate, Hokkaido 041–8655, Japan
a) y-nakamr@fun.ac.jp
b) osamu@fun.ac.jp

2. 関連研究

2.1 Concurrent Multi-path Transfer（CMT）

1台で複数のネットワークデバイスを持つ端末を称して，マルチホーム端末と呼ぶ．マルチホーム端末の end-to-end 間においてネットワークデバイスごとで複数の経路を結ぶことを，マルチパスと呼ぶ．Concurrent Multi-path Transfer（CMT）[4] とは，1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を行う技術である．CMT を用いることで，単位時間あたりの転送量を増加させることが可能となる．**図 1** では同種のパス2本を利用した CMT による転送と，シングルパスによる従来の転送を比較しており，CMT での転送はシングルパスによる転送に比べ単位時間あたりの転送量が2倍になっていることが分かる．CMT をアプリケーション層で実現する場合，アプリケーション開発時に CMT メソッドを組み込んで実現する．CMT 実現時の，各レイヤの構成を**図 2** に示す．アプリケーションは CMT 処理用のメソッドにデータを渡すことで相手先への転送を行う．CMT 処理用のメソッドは複数のソケットインタフェースを作成し，接続先と複数のコネクションを結び，アプリケーションから受けたデータの分割・送受信のスケジューリング・再結合を行うことで転送処理を行う．

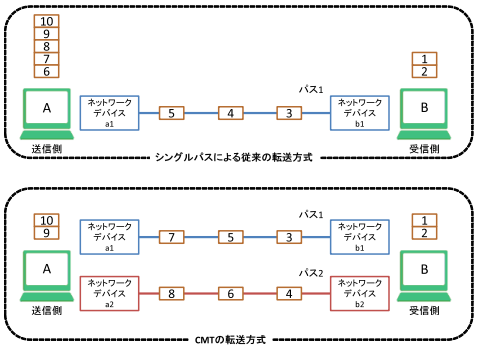

図 1 転送方式の比較
Fig. 1 Comparison of transmission technique.

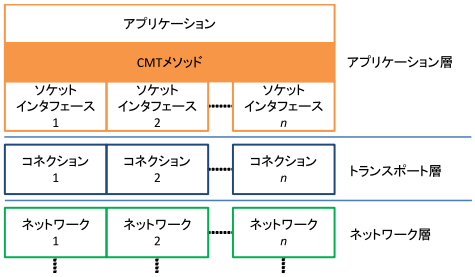

図 2 アプリケーション層での CMT の実現
Fig. 2 Realization of CMT in application layer.

2.2 GridFTP with Automatic Parallelism Tuning（GridFTP-APT）

GridFTP with Automatic Parallelism Tuning（GridFTP-APT）[5] とは，GridFTP [3] 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 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 コネクションごと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．ファイルの送信側・受信側で複数本の TCP コネクションを確立し，ファイルの断片（チャンク）をコネクションごとに並列に転送する．しかし，GridFTP には TCP コネクション数の明確な定義はないため，パラメータとして指定する必要がある．最適な TCP コネクション数はネットワークの環境によって定まるものであり，TCP コネクション数とスループットは単純に比例しないことが，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外的な要因でも変化するため，TCP コネクション数の最適値を事前に求めることは難しい．GridFTP-APT はこの問題に対して，転送中に都度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によって解決を試みている．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最適な 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 3** に示す．GridFTP-APT は図 3 の特性を利用し，チャンクを送信しながらスループットが最大となる最適 TCP コネクション数 N を特定している．GridFTP-APT アルゴリズムは主に 2 つのステップで最適な TCP コネクション数を探索する．ステップ 1 では，最適な並列 TCP コネクション数を含む範囲であるブラケット（ここでは 3 つの整数値で表す）を求める．ステップ 2 では，ステップ 1 で求めたブラケットに対し，黄金分割探索法 [8] を用い，最適 TCP コネクション数を特定する．

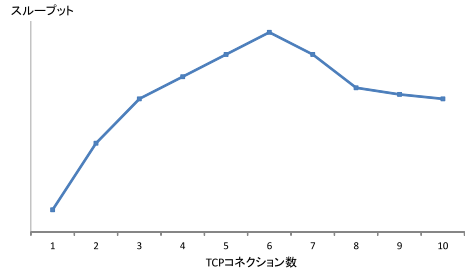

図 3 TCP コネクション数とスループットの関係
Fig. 3 Relations between the number of TCP connections and the throughput.

3. アドホック無線通信における高速なファイル転送方式

本論文では CMT による帯域幅の拡張と GridFTP-APT によるネットワークの転送効率増加を組み合わせた，隣接したマルチホーム端末どうし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．さらに，ネットワーク環境が変化しやすいアドホック無線通信環境に適した処理を取り入れた転送アルゴリズムを提案する．

3.1 提案方式の前提条件

提案方式では，アドホック無線通信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．本手法は隣接端末どうしで用いるため，無線ネットワークデバイスの個数は利用ユーザどうしでの確認が可能なためである．

(2) コネクション上限
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場合もある．

3.2 提案方式の概要

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．双方ともネットワークを介した通信を想定した技術であるが，提案方式はこれらの技術を 2 つの隣接端末機器間でのデータ交換に応用することで，高速なファイル転送を実現する．

送受信時におけるこれらの処理を行うクラスの関係を**図 4** 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 クラスが行い，送信の処理およびコネクション本数の調整処理を apt-send クラスが，受信処理を apt-recv クラスがそれ

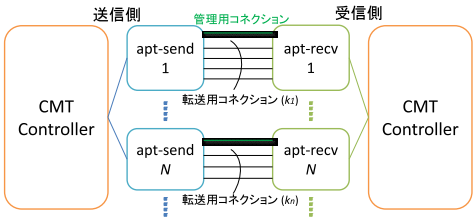

図 4 送信側と受信側の関係図

Fig. 4 Relationship diagram of the transmitting and receiving sides.

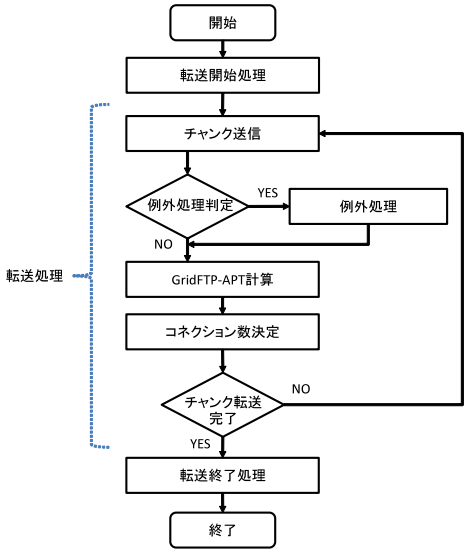

図 5 提案方式のフローチャート

Fig. 5 Flowchart of the proposed method.

ぞれ行う．apt-send および apt-recv 間では複数の TCP コネクションが確立されている．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．

提案方式のアルゴリズムは大きく分けて，転送開始処理，転送処理，転送終了処理の 3 フェーズで行われる．また，転送中にネットワーク環境が変化した場合を考慮し，例外処理と呼ぶフェーズを転送処理に組み込んでいる．提案方式のフローチャートを**図 5** に示す．ファイル転送を開始すると，まず転送開始処理が行われる．ここではチャンクの作成や，コネクションの確立処理などが行われる．転送処理では GridFTP-APT による最適 TCP コネクション数の計算を行いつつ，実際のファイル転送を行う．また，転送中にネットワーク環境の変化が起きスループットが低下した場合，例外処理が行われ GridFTP-APT の計算や転送用コネクション数の決定に影響を与える．転送処理を繰り返し，チャンクの転送が完了した後に，転送終了処理が行われる．ここでは最終的なチャンクの欠落確認やチャンクの結合処理などが行われ，終了後にファイル転送完了となる．

3.3 転送開始処理

転送開始処理は 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．しかし，分割後のチャンクの具体的なサイズは GridFTP-APT では定められておらず，最適なチャンクサイズを検討する必要がある．通信環境や使用する無線ネットワークデバイスの種類によって，最適なチャンクサイズは異なる可能性がある．

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 だけ同時並列に行い，チャンクをすべ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って 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，転送終了処理に移行する．

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た時点で，転送終了処理を開始する．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．すべ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，すべての 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 [10]．ネットワーク障害が発生し，転送中にコネクションが切断された場合は CMT Controller によって，管理用コネクションから未到着チャンクの再送を要求できる．パスそのものが切断された場合は，再度パスが接続された後，未到着のチャンクの転送処理に入る．ただし，隣接端末間での通信であるため，パスの切断はそれほど頻繁ではないうえ，スループットも低い状態となるため，張られるコネクション数も少なくなり，影響は小さい．またスループットの著しい低下が引き起こされる場合もある．このとき，マルチパスの利点を活かし，チャンクの割当てを柔軟にすることによって，ネットワーク障害発生時もデータ転送への影響を少なくすることができる．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．**図 6** に apt-send の例外処理判定のフローチャートを示す．CMT Controller は各ネットワークデバイスのスループットを比較しながら監視する．ネットワークデバイス間でスループットに大きな差異が認められた場合，すなわち著しく低いスループットを持つ遅いネットワークデバイスの存在が観測された場合，ネットワークの障害発生と認識し，遅いネットワーク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．このとき，スループットが 1/2 であるようなコネクションは，同等の性能を持つコネクションの障害発生時と同程度に遅いコネクションであるといえるため，提案方式では最も高いスループットと比較して 1/2 を下回るスループットを著しく低いスループットの基準とする．これらの例外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割り当てることにより，ネットワーク障害発生時の転送への影響を低減させている．

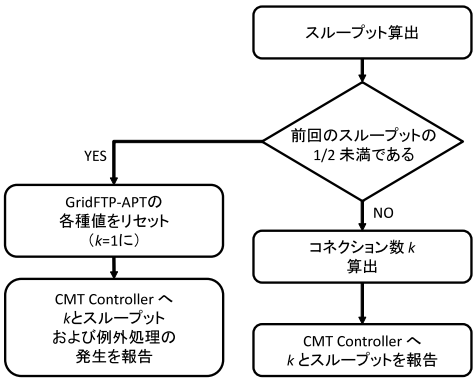

図 6 例外処理：apt-send の振舞い
Fig. 6 Exception handling: The behavior of apt-send.

4. 評価実験

提案方式を PC 上に仮想的に実装し，性能の評価を行う．また，転送中に意図的にネットワークに障害を起こすことで提案方式の例外処理を動作させ，その有効性についても評価を行う．

4.1 実験環境

アドホック無線通信は外的要因による通信状態の変化が激しく，周囲の電波環境によらない環境での評価が難しい．そこで，仮想マシン/ネットワークを用いた実験環境を用意し，提案方式の実装・評価実験を行う．仮想化には VMware player [11] を用い，**表 1** のスペックの仮想マシンを 2 台作成した．

4.2 正常転送時の評価

提案方式の有効性を確かめるため，従来の転送方式である FTP，マルチパス・シングルコネクションの CMT，マルチパス・マルチコネクションである提案方式の 3 種で転送実験を行い，各方式別の平均スループットを求めて比較・評価した．ここではネットワーク障害などの外的要因により評価が混乱することを避けるため，ネットワーク遅延やパケットロスなどがいっさい起こらない環境で実験を行った．実験では 1 Gbyte のファイル転送を行い，転送に

表 1 実験用仮想マシンのスペック
Table 1 Specifications of the virtual machines.

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

表 2 正常転送時の平均スループット
Table 2 Average throughput in the normal case.

	平均スループット (Mbps)
FTP	10.2
CMT	20.3
提案方式	20.4

要した時間からスループットを求める．FTP デーモンには vsftpd [17] を用いている．また，CMT および提案方式で用いるチャンクサイズは，元ファイルの 1/100 である 10 Mbyte で固定し実験を行っている．

各方式別の結果を**表 2** に示す．この実験環境では外部から通信に与える影響が存在しないため，提案方式の Grid-FTP-APT 部分は最もスループットが高くなるまでコネクション数を増加させるのみであり，最適なコネクション数に落ち着いたあとは従来の CMT とほぼ同じ処理を行っている．最適コネクション数に達するまでの間は CMT よりスループットが低下している可能性があるが，表 2 より転送終了時には CMT とほぼ同等の平均スループットを達成できており，この問題は転送効率に大きな影響を与えていない．また，FTP に比べ，CMT および提案方式は平均スループットが 2 倍近くになっている．これは CMT および提案方式ではネットワークデバイスを 2 つ用いているため，妥当な結果である．

4.3 例外処理発生時の評価

提案方式では，外的要因による通信状態の変化が激しいアドホック無線通信に対応するため，例外処理アルゴリズムを組み込んでいる．この例外処理の有効性を確かめる評価実験を行う．例外処理はマルチパスを前提にしているため，CMT と提案方式についてのみ比較評価を行っている．

4.3.1 実験シナリオ

意図的に例外処理を発生させるため，ネットワークエミュレータを用い転送中にネットワークの環境を変化させる．ネットワークエミュレータには netem [12] を用いる．正常時の転送にかかる時間のおよそ半分である 180 sec 後に，ネットワーク 2 に障害を発生させ，障害発生後もネットワーク 2 は完全には遮断されず，送受信は可能ではあるがスループットが約 1/25 まで低減する，というシナリオで実験を行う．これにより，スループット低下を検知した際の apt-send の振舞い，ネットワークデバイス間の速度差を検知した際の CMT Controller の振舞いを検証する．実験では netem を用い，転送開始 180 sec 後から転送終了までの間，ネットワーク 2 に対して 3000 msec の遅延をかけ続けることで，ネットワーク 2 のスループットを約 1/25，すなわち 0.5 Mbps まで低下させる通信環境を実現する．**図 7** にネットワーク 2 のスループット変動の様子を示す．このもとで 4.2 節と同様に，1 Gbyte のファイル転送を行い，転送に要した時間からスループットを求める．このときのチャンクサイズは，元ファイルの 1/100 である 10 Mbyte で固定する．

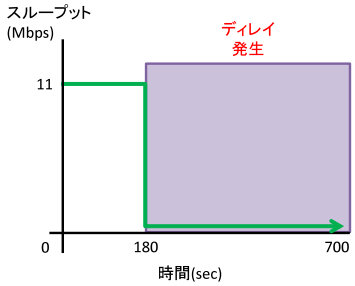

図 7 スループット変動シナリオ
Fig. 7 Throughput variation scenario.

表 3 例外処理発生時の平均スループット
Table 3 Average throughput in the exception case.

	平均スループット (Mbps)
CMT	2.8
提案方式（例外処理無効）	11.6
提案方式（例外処理有効）	13.6

表 4 チャンクの個数
Table 4 Number of chunks.

	デバイス 1	デバイス 2
CMT	50	50
提案方式（例外処理無効）	62	38
提案方式（例外処理有効）	71	29

4.3.2 実験結果

転送中にスループット変動が起きても再スケジューリングなどの対応を行わない CMT と，例外処理を無効にした提案方式，有効にした本来の提案方式の 3 種について平均スループットを求め，各方式別の結果を**表 3** に示す．実験結果より，転送中にスループット変動が起こった際に，提案方式は CMT に比べ，5 倍近い平均スループットを維持している．

次に，各デバイスで最終的に送信したチャンクの個数を**表 4** に示す．スループットの変動時に再スケジューリングを行わない CMT では，各スループットに等しくチャンクを割り当てていることが分かる．スループットが転送中に急激に低下するネットワーク 2 の影響を受けたことが，CMT の平均スループットが大きく低下させた原因であると考えられる．

次に，例外処理の有効性について考察する．例外処理を無効にした提案方式と，有効にした提案方式の平均スループットを比較すると，有効にした場合が 20%程度向上している．この要因を明らかにするため，例外処理無効時と有効時それぞれのコネクション数の時間あたりの推移を表したグラフを**図 8** と**図 9** に示す．各グラフにプロットされた点は，その時刻に送ったチャンクの個数を表している．

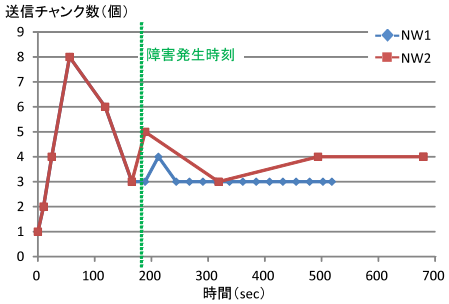

図 8 コネクション数推移（例外処理無効の場合）

Fig. 8 Variations of the number of connections (Exception processing is disabled).

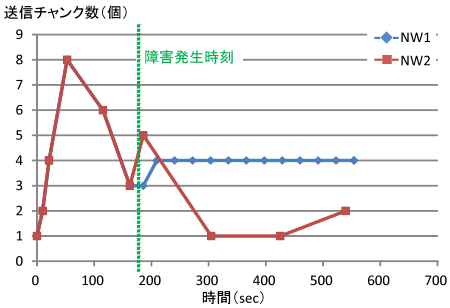

図 9 コネクション数推移（例外処理有効の場合）

Fig. 9 Variations of the number of connections (Exception processing is enabled).

0 秒から 180 秒まではスループット変動がないため正常転送時と同様に，チャンクの送信のたびにコネクション数を倍加させていき，最適なコネクション数を超えたあと，最適なコネクション数までコネクションを減らすと処理を行っており，送信チャンク数はコネクション数の増減にともなって変化している．図 8 から，例外処理を無効にした場合は，ネットワーク 2 の障害発生後も，コネクション数を減らすことなく送信を続けていることが分かる．しかし障害発生後からはネットワーク 2 でのプロットの間隔が広くなっており，チャンク割当ての再スケジューリングが働いていることが分かる．このため，多くのチャンクを遅延の大きいコネクションを用いて転送することになり，スループットの低下につながっている．図 9 では例外処理の働きが見られる．ネットワーク 2 の障害発生後，プロットの間隔が広くなっているものの，コネクション数も急激に減少している．例外処理を有効にした場合は，最終的に正常なネットワーク 1 から送信したチャンクが多く，障害が発生したネットワーク 2 からの送信チャンクが少なくなっており，より効率良くコネクションに割り当てられていることが表 3 の結果から分かる．

4.4 最適なチャンクサイズの評価

GridFTP-APT アルゴリズムはファイルを分割しチャンクを作成するが，3.3 節で述べたように，チャンクサイズについては明確に定められていない．本節では提案方式が想定する環境において，正常時と例外処理発生時，両方の場合について最適なチャンクサイズを評価する．いずれの場合においても最適なチャンクサイズは x Mbyte と一意には定められない可能性を考慮するため，提案方式ではチャンクの分割割合に加え，トータルのファイルサイズも考慮している．

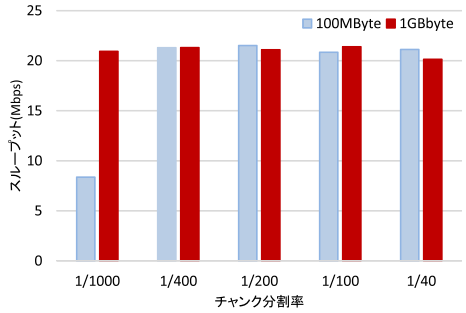

図 10 チャンク分割率別のスループット

Fig. 10 Throughput according to the chunk division ratio.

表 5 チャンクサイズ別のスループット

Table 5 Throughput according to the chunk size.

	チャンクサイズ 100 kbyte	チャンクサイズ 1 Mbyte
100 Mbyte	8.36	20.84
1 Gbyte	8.23	20.93

4.4.1 正常時における最適チャンクサイズ

本項では，ネットワーク遅延やパケットロスなどがいっさい起こらない場合における最適なチャンクサイズを求める．100 Mbyte と 1 Gbyte のファイルをそれぞれ 1/1000〜1/40 の間で 5 段階に分割してチャンクを作成し，転送実験を行う．転送にかかった時間から平均スループットを求め，チャンクサイズによるスループットの変化を調べる．**図 10** に結果を示す．分割率 1/400〜1/40 の間では，100 Mbyte，1 Gbyte ともに大きな平均スループットの変化はないが，100 Mbyte の分割率 1/1000 の場合のみ，平均スループットが大きく低下していることが分かる．1 Gbyte の分割率 1/1000 では平均スループットの急激な低下は見てとれないため，チャンクの個数が多すぎるという問題ではなく，チャンクのサイズによるものと考えられる．100 Mbyte の分割率 1/1000 の場合，実際のチャンクのサイズは 100 kbyte となる．そこで 1 Gbyte をチャンクサイズが 100 kbyte となるよう，1/10000 に分割して送った場合の平均スループットを測定した．測定結果を**表 5** に示す．チャンクサイズ 100 kbyte で送った場合，1 Gbyte の場合でも 100 Mbyte と同様に，平均スループットが急激に低下していることが分かる．参考として表 5 にはチャンクサイズ 1 Mbyte（100 Mbyte の場合は分割率 1/100，1 GB の場合は分割率 1/1000）の場合の平均スループットを載せているが，チャンクサイズ 100 kbyte の場合と比べ 2.5 倍ほどのスループットが出ていることが分かる．

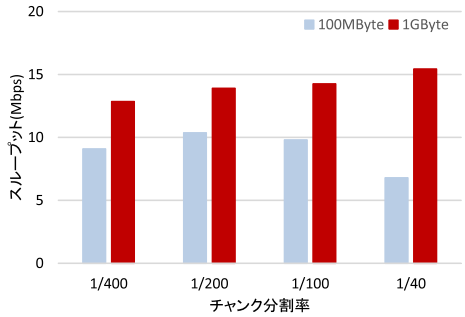

図 11 チャンク分割率別スループット（パケットロス非発生時）
Fig. 11 Throughput according to the chunk division ratio (without packet loss).

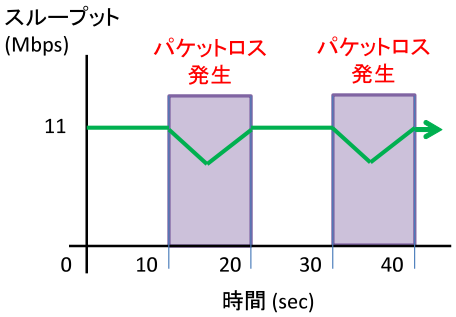

図 12 ネットワーク 2 のスループット変動シナリオ
Fig. 12 Throughput variation scenario of network 2.

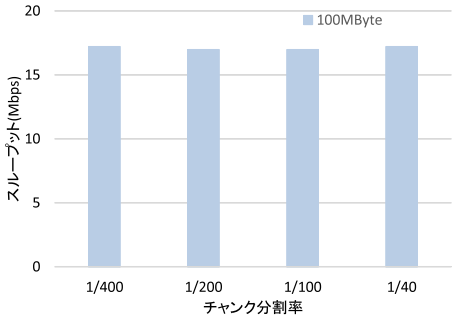

図 13 チャンク分割率別スループット（パケットロス発生時）
Fig. 13 Throughput according to the chunk division ratio (with packet loss).

4.4.2 例外処理発生時における最適チャンクサイズ

4.4.2.1 パケットロスが発生していない場合

例外処理発生時における平均スループットの変化をチャンクサイズ別に調査する．前項の結果をふまえて，100 Mbyte と 1 Gbyte のファイルをそれぞれ 1/400〜1/40 の間で 4 段階に分割し，チャンクを作成して転送実験を行う．なお，実験は 4.3.1 項と同じシナリオを利用する．結果を**図 11** に示す．100 Mbyte のファイル転送時，1/400〜1/40 の 4 段階の間で凸型を描く結果となった．しかし 1 Gbyte のファイル転送時では，右上がりに平均スループットが向上している．これにより，最適と思われるチャンクサイズの条件は，チャンク分割率や障害の発生条件が等しい場合でも，トータルのファイルサイズによって異なることが分かる．ある程度のスループットが確保されている通信では，基本的にチャンク分割率が小さい方がスループットは向上する．一方チャンク分割率が大きい場合は，4.4.1 項と同様，チャンクサイズが 250 kbyte を下回ったときにはスループットが低下するが，ネットワーク障害が発生し，利用可能な帯域が減少している場合には，サイズの小さいチャンクの通信しかできない場合もあり，このときにはスループットがやや向上する．

4.4.2.2 パケットロスが発生している場合

4.4.2.1 と異なり，あるコネクションのスループットに頻繁な増減がみられる場合のチャンクサイズが全体のスループットに与える影響を見るため，例外処理を発生させる障害の条件を変え，転送中に一定のタイミングでパケットロスが発生し，一時的にスループットが低下する，という条件のもと実験を行う．実験のシナリオを**図 12** に示す．転送開始から 10 sec〜20 sec，30 sec〜40 sec の間は 30%の確率でネットワーク 2 にパケットロスが発生し，平均スループットがが 30%程度低下する，というシナリオのもと，100 Mbyte のファイルをそれぞれ 1/400〜1/40 の間で 4 段階に分割し，チャンクを作成して転送実験を行う．**図 13** の実験結果からはチャンクサイズによって平均スループットに目立った差異は生じなかった．この結果から，同サイズのファイルであっても，障害の条件によって最適と思われるチャンクサイズの条件は異なることが分かる．提案方式では，転送開始時にチャンクサイズを決定して分割しているが，利用する無線ネットワークデバイスの種類や転送中の環境によって，動的にチャンクサイズを変化させる方式が有効となる場合も考えられる．

5. おわりに

本論文では，モバイル端末間におけるアドホック無線通信を用いた高速なファイル転送方式を提案することを目的とし，転送速度を高速化させる手法として，マルチパス手法をとる CMT とマルチコネクション手法をとる GridFTP-APT という 2 つの技術を組み合わせたファイル転送方式を提案した．さらに，アドホック無線通信の特徴としてスループットの変動など通信環境の変化が激しい点に着目し，その回避策であるマルチパス・マルチコネクションを活かした例外処理を考案し，提案方式の実装を行った．転送中のネットワーク障害が発生しない場合，転送中にネットワーク障害が発生する場合，それぞれについて従来手法との比較・評価を行い，転送中の平均スループットについて提案方式の優位性を示すことができた．

今後は様々なネットワーク環境や無線ネットワークデバイスの組合せに対応できるよう，チャンクサイズの動的変化に対応した手法の検討が必要であると考えられる．

参考文献

[1] Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications, IEEE Std 802.3, 2000 Edition (2000).

[2] Altman, E. and Barman, D.: Parallel tcp sockets: Simple model, throughput and validation, *Proc. 25th IEEE International Conference on Computer Communications* (*INFOCOM2006*), pp.1–12 (2006).

[3] GridFTP v2 Protocol Description, available from ⟨http://www.ogf.org/documents/GFD.47.pdf⟩ (accessed 2015-05-14).

[4] Iyengar, J.R., Amer, P.D. and Stewart, R.: Concurrent Multipath Transfer Using Transport Layer Multihoming: Performance Under Varying Bandwidth Proportions, *Proc. 2004 IEEE Military Communications Conference* (*MILCOM 2004*), Vol.1, pp.238–244 (2004).

[5] 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，Vol.105, No.472, pp.19–24 (2005).

[6] The Globus Alliance, available from ⟨http://www.globus.org/⟩ (accessed 2015-05-14).

[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，Vol.104, No.182, pp.19–24 (2004).

[8] Press, W.H.: *Numerical Recipes in C : The Art of Scientific Computing*, Cambridge University Press (1992).

[9] Ito, T., Ohsaki, H., Imase, M.: GridFTP-APT: Automatic Parallelism Tuning Mechanism for GridFTP in Long-Fat Networks, *Proc. 7th Asia-Pacific Symposium on Information and Telecommunication Technologies* (*APSITT2008*), pp.189–194 (2008).

[10] 安藤康臣，平岡精一：不安定な無線通信環境における階層型省電力プロトコルに関する検討，電子情報通信学会技術研究報告，Vol.107, No.447, pp.31–35 (2008).

[11] VMware, available from ⟨http://www.vmware.com/⟩ (accessed 2015-05-14).

[12] netem, available from ⟨http://www.linuxfoundation.org/⟩ (accessed 2015-05-14).

[13] 長谷川剛，村田正幸：TCP の輻輳制御機構に関する研究動向，電子情報通信学会論文誌，Vol.J94-B, No.5, pp.663–1672 (2011).

[14] Stevens, W.R.: TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms, Internet RFC 2001 (1997).

[15] Stewart, R.: Stream Control Transmission Protocol, Internet RFC 2960 (2000).

[16] 佐竹伸介，稲井　寛，齊藤智也，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，システム制御情報学会論文誌，Vol.25, No.2, pp.19–27 (2012).

[17] vsftpd, available from ⟨http://vsftpd.beasts.org/⟩ (accessed 2015-05-14).

中村 嘉隆 （正会員）

2002 年大阪大学基礎工学部情報科学科卒業．2007 年同大学大学院情報科学研究科博士後期課程修了．同年奈良先端科学技術大学院大学情報科学研究科助教．2010 年大阪大学大学院情報科学研究科特任助教．2011 年より公立はこだて未来大学システム情報科学部助教．博士（情報科学）．ユビキタスネットワークに関する研究に従事．電子情報通信学会，IEEE 各会員．

高橋 大斗

2012 年公立はこだて未来大学システム情報科学部情報アーキテクチャ学科卒業．2014 年同大学大学院システム情報科学研究科システム情報科学専攻博士前期課程修了．在学中はアドホック無線通信に関する研究に従事．現在，ドコモ・テクノロジ株式会社勤務．

高橋 修 （正会員）

1975 年北海道大学大学院工学研究科修士課程修了．同年日本電信電話公社（現，NTT）横須賀電気通信研究所入所．コンピュータネットワークの研究・開発・標準化に従事．NTT ドコモを経て，2004 年より公立はこだて未来大学教授．博士（工学）．本会業績賞．電子情報通信学会，IEEE 各会員．本会フェロー．