

近距離アドホック無線通信における 高速/高信頼性トランスポートプロトコルの提案と実装

高橋 大斗[†] 中村 嘉隆[†] 白石 陽[†] 高橋 修[†]

公立はこだて未来大学 システム情報科学部[†]

1 はじめに

近年、携帯端末などの高機能化により、隣接した端末間での、アドホック無線通信による直接のデータ交換の頻度は高まっている。しかし、端末の高機能化に伴い扱うデータの量は増える傾向であるのに対し、データの転送について十分な速度を実現できていない。データ転送の高速化についての手法は多々あるが、ここではトランスポート層における高速化について考える。高速転送を目的とするトランスポートプロトコルとしては UDP (User Datagram Protocol) が一般的であるが、TCP (Transmission Control Protocol) に比べて信頼性に劣るため、アプリケーション開発時には大きな負担となる。端末の高機能化に伴って、信頼性の高いかつ高速なトランスポートプロトコルの需要は、今後一層高まると考えられる。

本研究では、この需要に答えることのできる、新しいトランスポートプロトコルを提案したい。端末の高機能により、一つの端末上に複数のネットワークデバイスを持つ、マルチホーム端末も珍しくは無くなった。マルチホーム端末においてパケットの並列同時転送を行う CMT (Concurrent Multipath Transfer) [1] という技術を用い、隣接した端末間でのアドホック無線通信による、高信頼/高速トランスポートプロトコルを実装することを、本研究の目的とする。

2 関連技術

2.1 CMT

マルチホーム端末の end-end 間において、ネットワークデバイス別に複数の経路を結ぶことを、マルチパスと呼ぶ。マルチパスを利用して、一つのデータを複数の経路で並列同時に転送することを、CMT と呼ぶ。

アドホック無線通信環境におけるマルチパス通信には、経路選択における問題など、実現を困難とする要素が多々存在することが関連研究

から明らかになっている[2]。しかし本研究では近接した端末間の、1hop でのデータ転送を対象としているため、これらの問題は起きないという前提で議論を進める。

2.2 SCTP

SCTP (Stream Control Transmission Protocol) [3] は TCP とほぼ同等の順番制御、エラー制御機能を持つ、コネクション指向型のトランスポートプロトコルである。本研究では SCTP の信頼性もさることながら、その付加機能に着目し、ベースとなるプロトコルとして採用した。SCTP は 2000 年に RFC に採用された比較的新しいプロトコルであるため、TCP には無い様々な付加機能を持っており、その一つにマルチホーム機能がある。

SCTP におけるマルチホーム機能とは、複数のネットワークデバイスをもつ一つのトランスポート層インタフェース (Linux におけるソケットインタフェース) で扱う機能である。現行の SCTP におけるマルチホーム機能は、複数登録されたネットワークデバイスの中からプライマリだけを選択してデータ転送に使用し、それ以外のデバイスは障害発生時のバックアップ用途に利用されるのみとなっている。SCTP におけるマルチホーム機能を改良し、ネットワークデバイスの複数同時利用を可能にすることで、CMT は実現可能であると考えられる。

3 提案手法

3.1 CMT の実現方式

本研究では、既存のプロトコルである SCTP を改良し、その信頼性を損なわず、CMT によるデータ転送高速化の実現方法を提案する。図 1 に CMT のフローを示す。

転送データの分割・結合の処理を行う中間層を、アプリケーション層とトランスポート層の間に設け、転送処理そのものはトランスポート層で行うこととする。提案方式では、NIC (Network Interface Card) 間の 2 つのパスそれぞれについてバッファを用意し、個別に輻輳制御、再送制御を行う。パス別に独立した動作を行うことで、内部制御を簡素化して信頼性を損なわ

“Proposal and implementation of fast and reliable transport protocol in ad-hoc wireless networks”

Hiroto Takahashi[†], Yoshitaka Nakamura[†], Yoh Shiraiishi[†], Osamu Takahashi[†]

[†]School of Systems Information Science, Future University Hakodate

ずに CMT を行うためである。

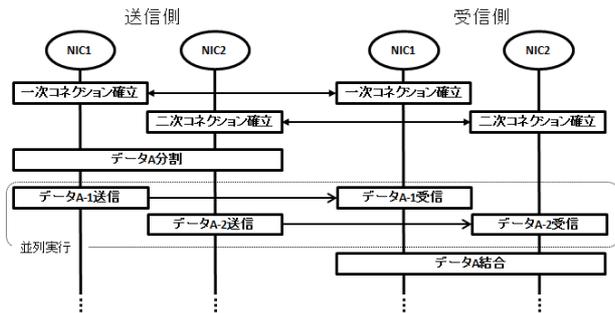


図 1 CMT 概要

また、CMT ではパス別に異なったシーケンス番号の packets が送られるため、受信後に再度シーケンス番号に従った並べ替えが必要になる。提案方式では図 2 に示すように、パス別に再度シーケンス番号を設定する階層型シーケンス制御を導入し、この問題を解決する。



図 2 階層型シーケンス制御

3.2 実装方法

SCTP は現状、Linux カーネル 2.6 以降に搭載されている。本研究では、カーネル 2.6 を用いた OS である Ubuntu11.04 を用い、この上に、SCTP ソケットライブラリである lksctp[4] をインストールし、SCTP を利用するための環境を整えた。lksctp はオープンソースのライブラリであり、このライブラリを改変することで、提案方式の実装を進める。

4 予備実験

4.1 実験環境

表 1 に、実験環境を示す。同様の仕様を持つ端末を 2 台用意し、端末間の転送速度を比較する。

表 1 実験用端末の仕様

CPU	Pentium M 1.1GHz
メモリ	768MB
OS	Ubuntu11.04
NIC1	Intel PRO/Wireless 2200BG
NIC2	BUFFALO WLI-UC-GN

4.2 並列転送の実験結果

予備実験用のプロトタイプとして、分割・結合・2つのパスによる並列転送の処理を、アプリケーション層で一括して行うプログラムを作成した。

このプロトタイプを使用し、予備実験として

通常の Single pass による転送方式を用いたプログラムとの転送時間の比較を行った。どちらも転送に用いるプロトコルは SCTP である。

転送するデータは 5MB・10MB・20MB のバイナリファイルとし、2つの方式につき前節の環境において各 10 回ずつ転送を行い、転送にかかった時間の平均を比較した。結果を図 3 に示す。

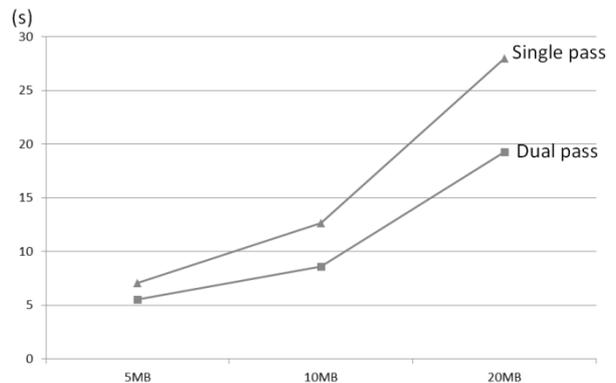


図 3 実験結果

4.3 考察

予備実験より、提案方式のプロトタイプによる並列 (Dual pass) 転送は、従来方式の Single pass による転送に比べて転送時間を短縮することができた。これは転送速度について、並列転送の優位性を示す結果と言える。

5 おわりに

本研究では、1hop でのアドホック無線通信における転送速度向上、および信頼性維持を目的とし、既存トランスポートプロトコルである SCTP の改良による実装方法について考察した。予備実験の結果より、提案方式を用いることでの転送速度向上は明らかとなった。

今後は提案方式に則った上で、トランスポート層および副層で処理を完了できるように、実装を進めたいと考える。

参考文献

- [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] アドホックネットワークにおけるマルチパストランスポートプロトコル, Mehdad N. Shirazi, 内藤壮司, 美濃 導彦, 情報処理学会研究報告. UBI, 2005(28), pp.99-103, 2005.
- [3] L. Ong, ed, "An Introduction to the Stream Control Transmission Protocol (SCTP)," RFC3286, 2002
- [4] Linux Kernel Stream Control Transmission Protocol (lksctp) project, <http://lksctp.sourceforge.net/>