

アドホックネットワークにおける中継遅延の削減方式

A Method of Decreasing Delay of Relay in Ad-hoc Networks

龍本 栄二[†] 滝沢 泰久[†] 門脇 直人[†] 小花 貞夫[†] 大久保 英嗣[‡]
 Eiji Takimoto Yasuhisa Takizawa Naoto Kadowaki Sadao Obana
 Eiji Okubo

1. はじめに

アドホックネットワークは、ネットワークを構成するすべてのノードがパケット中継機能を持つため、インフラストラクチャを用いないネットワーク構築が可能であるという利点を持つ。このことから、アドホックネットワークは、様々な用途への適用が期待されている。アドホックネットワークの適用アプリケーションの1つに、ITS(高度交通システム)における車車間通信がある。車車間通信は、各車両が1つの通信ノードとしての役割を持ち、互いの位置情報や危険情報といった車両に関する情報を交換することで安全性を向上させることを目的としている。このようなアプリケーションでは、発生したイベントをリアルタイムに周辺車両へ通知することが必要となる。しかし、マルチホップ通信では、中継処理による遅延が生じるため、高いリアルタイム性の実現が困難である。

そこで、我々は、アドホックネットワークにおける中継処理の遅延を削減する手法を検討した。本手法では、車両情報をブロードキャストによるフラッディングで、周辺車両に通知することを想定している。本手法は、パケット中継時に各ノードにおいて行われるソフトウェア処理[§]を対象とし、その処理を下位層であるデバイスドライバで行うことで遅延削減を実現している。同様に、パケットの送受信をアプリケーションとデバイスドライバ間で直接行うことにより、送受信の高速化も実現している。

以下、本稿では、2章で提案手法の概要、3章で性能評価について述べ、4章で本稿をまとめについて述べる。

2. 提案手法の概要

パケットの中継を行う上で生じる遅延は、大きく分けて2種類ある。1つは、遅延処理による遅延である。遅延処理は、OSの処理効率向上のために広く使われている処理方式である。通信では、複数のパケットをまとめて処理するために遅延処理が用いられている。高速化を実現するためには、パケットを受信したら即座に処理を行う必要がある。

もう1つの遅延は、コンテキストスイッチによる遅延である。フラッディングの最も容易な実装方式は、アプリケーションレベルでの実装である。この実装では、パケット受信毎にフラッディングの処理を行うアプリケーションを実行する必要がある。このとき、アプリケーションを呼び出すためにコンテキストスイッチが生じる。

[†](株) 国際電気通信基礎技術研究所 適応コミュニケーション研究所, Advance Telecommunications Research Institute International, Adaptive Communication Research Laboratories

[‡]立命館大学情報理工学部, Ritsumeikan University College of Information Science and Engineering

[§]本稿では、遅延とはこの処理を指すものとする。

ID	SEQ	TTL	LEN	DATA
(8)	(8)	(8)	(16)	(0-2307Bytes)

ID(8bits):Host Identify Number
 SEQ(8bits):Sequential Number
 TTL(8bits):Time To Live
 LEN(16bits):Data Length(Bytes)

図1: フラッディング用ヘッダ

提案手法では、送受信処理と中継処理というフラッディングに関わる処理を、すべてデバイスドライバで行う。これにより、上記の遅延を削減し、中継処理を高速化している。以下、本章では、各処理における高速化手法について述べる。

2.1 中継処理の高速化

IPを用いたブロードキャストでは、IPアドレスにブロードキャストアドレスを用いる。このとき、MACフレームの宛先MACアドレスにもブロードキャストアドレスが用いられる。このことから、MACフレームのみを参照すれば、そのパケットがフラッディングによるものか否かを判断することが可能である。そこで、提案手法では、IPを用いず、図1に示す簡素なフラッディング用ヘッダを用いてフラッディングを実現している。

IDは、ノード識別子である。本手法では、ノードに割り当てられたIPアドレスの下位8ビットをIDとして用いるものとした。SEQは、パケットに与えられる番号であり、ノードが新たにパケットを生成するたびに1つ値が加算される。パケットの中継処理において、既知のパケットか否かを判断するために用いられる。TTLは、パケットの生存時間を表し、1回中継される毎に1つずつ値が減算される。TTLを用いることで、フラッディングの到達範囲を指定することが可能である。例えば、Helloパケットの場合は、TTLを1と指定することで実現することができる。LENは、フラッディングヘッダ長を除くパケットのデータ長を表す。DATAは、データ本体である。

本手法では、フラッディングアルゴリズムにシンプルフラッディングを採用している。そのため、フラッディング用ヘッダのIDとSEQをキャッシュすることで、冗長なパケット中継を回避している。本手法の処理手順は、以下に示す通りである。

手順1. フラッディングパケットか否かの判断

- フラッディングパケットであれば、以下の手順を行う。そうでなければ、通常の処理(上位レイヤへパケットを渡す)を行う。

手順2. 冗長性のチェック

- はじめて受け取るパケットであれば手順3を行い、キャッシュを更新する。
- 冗長パケットであればパケットを破棄して終了する。

手順3. MACヘッダとフラッディング用ヘッダ書換え
 手順4. 受信キューへの挿入と送信処理

以上の処理は、すべてデバイスドライバ内で行われる。また、応答性を高めるために、割込みコンテキスト内で処理を完了させる。これにより、他のカーネル内処理によるブロッキング時間を最小化している。

2.2 送受信の高速化

通信は、対象となるノードすべてがパケットを受信し、アプリケーションに渡されて完了する。すなわち、中継処理を高速化だけでなく、アプリケーションの送受信も高速化することが必要であると考えられる。そこで、本手法では、アプリケーションがデバイスドライバに対して直接送受信要求を出すことで、送受信の高速化を行う。これにより、送受信処理をソケットライブラリやTCP/IP層を含まないため、より高速な通信が可能になる。

2.3 プロトタイプシステム

我々は、提案手法に基づき、プロトタイプシステムの実装を行った。プロトタイプシステムのシステム構成は、以下に示す通りである。

- 計算機：CPU PentiumIII プロセッサ 700MHz
メモリ 512~768M バイト
- 通信デバイス：Prism3 チップ使用 IEEE802.11b 無線 LAN カード
- OS：FedoraCore3 Linux(kernel2.6.9)
- デバイスドライバ：linux-wlan-ng0.2.3

Linux は、オープンソースであることと、サーバから組み込み用途まで多様な使用実績があることから採用した。通信デバイスには、一般的であり入手も容易であることから IEEE802.11b デバイスを採用した。デバイスドライバは、Linux と Prism3 チップの組合せにおいて使用実績がある linux-wlan-ng を用いた。

図2に、プロトタイプシステムの構成を示す。デバイスドライバは、受信パケットがブロードキャストであれば、キャッシュを参照し、初めて受信したパケットのみを中継し、かつ受信キューに格納する。キャッシュは、図1のIDとSEQの組み合わせで管理される。アプリケーションとデバイスドライバとのインタフェースは、実装の容易な ioctl システムコールを用いた。プロトタイプでは、ioctl システムコールに、フラッディングパケットの送受信用インタフェースを加えた。

3. 性能評価

実機上に実装したプロトタイプシステムを用い、その性能の評価を行った。実験では、計算機を直線に配置し、両端をそれぞれ送信ノードと受信ノードとし、中間に配置されたノードを中継ノードとし、このトポロジ上で、ホップ数を1~5ホップまで変化させて行った。評価は、送信ノードがパケットを送信して、受信ノードが受信するまでの時間を計測した。ただし、全ノードの時刻を正確に合わせることが困難であるため、パケットの往復にかかった時間を2で割った値を結果として用いた。また、パケットサイズは、100バイトとした。また、比較対象となるフラッディングアプリケーションを2つ用意した。これらのアプリケーションは、ソケット作成時にそれぞ

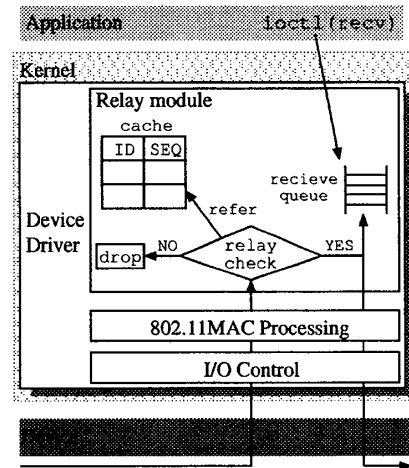


図2: プロトタイプシステムの概要

れ SOCK_RAW オプションと SOCK_DGRAM オプションを用いている。これらでは、フラッディングはソケットを通してアプリケーションで処理される。

実験の結果を図3に示す。このことから、提案手法は、他の手法に比べ、1中継ノード当りの遅延を約33%軽減することができた。削減された遅延の内訳は、遅延処理に関するものが55%、コンテキストスイッチに関するものが42%であった。残りの3%は、ヘッダの処理等に関するものであった。また、中継ノードを挟まない1ホップの場合も、提案手法が低遅延であることから、送受信の高速化が、通信の高速化に効果的であることが分かる。

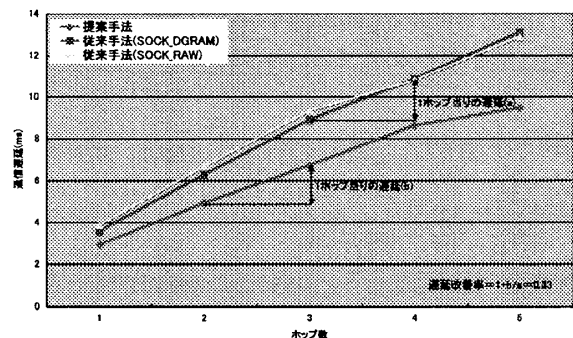


図3: 実験結果

4. おわりに

本稿では、アドホックネットワークにおけるフラッディングの中継遅延を削減する手法について述べた。本手法は、デバイスドライバ内で中継処理を行うことにより、遅延の削減を実現している。また、プロトタイプシステムを実機上に実装し、中継遅延を約33%削減されることを確認した。今後は、フラッディングだけでなくユニキャスト通信における中継遅延の削減に関する手法についても検討を行う予定である。

謝辞

本研究は、情報通信研究機構の研究依託「高レスポンスマルチホップ自律無線通信システムの研究開発」により実施した。