

通信プロトコルのモバイルコード化による 無線ネットワークシミュレーション環境の構築

御子柴 健吾 伊藤 洋 大坐島 智 川島 幸之助

東京農工大学

1. はじめに

無線ネットワークのシミュレーションにおいて、無線の特性を忠実に模擬することは極めて難しい。近年、これを解決するために、無線部分に実機を用いたテストベッドが開発されている。実機の使用により、現実に近い無線ネットワークシミュレーションを実現している。

その場合、シミュレーションで使う実機ノード全てにプロトコルを、その都度インストールする必要がある。しかも、アプリケーション層より下位層のプロトコルはカーネル内部のプログラムであり、通常、新たなプロトコルの導入にはカーネルの再構築を行わなければならない。

一方、ネットワークを介してプロトコルを自動的に取得し実行可能にするモバイルコードが、近年注目を浴びている。これにより動的にシステムの変更・拡張ができ、プロトコルの普及手段として有用である。

本稿では、トランスポート層、IP層のプロトコルをモバイルコード化してPC端末に配布し、アドホックネットワークプロトコルの性能評価ができる環境を構築する。

2. システム設計

プロトコルを評価するために、無線環境におけるシミュレーション実験を自動化し、サーバで管理できるようにする。プロトコルのモバイルコードは、サーバに保管されている。クライアントは必要時に、サーバからファイルをダウンロード・解凍・コンパイルして実行する。

2.1 プロトコル配布方法

モバイルコードにより、トランスポート層で使用するTCP、UDPプロトコルを動的に変更できるシステムSTP[1]を改造し、モバイルコードでトランスポート層、IP層のプロトコルを実装する。これにより、カーネルの再構築無しに新し

いプロトコルで動的に構成可能となる。

サーバ側にはプロトコルがハッシュ値をファイル名にしたアーカイブファイルとしてサーバに配置される。クライアント側では、バックグラウンドでプロトコルローダを起動している。ローダは、デバイス部とシェル部からなり、デバイス部がハッシュ値としてプロトコル要求を受信し、シェル部に送信し、シェル部がサーバから該当ファイルをダウンロードする。

2.2 プロトコルのモバイルコード化

ユーザプロセスからトランスポート層へ、ネットワークインタフェースからIP層へのアクセスを定義しているポインタがカーネル内部を指している。ポインタがモバイルコード利用時にモバイルコード内を指すようにした。

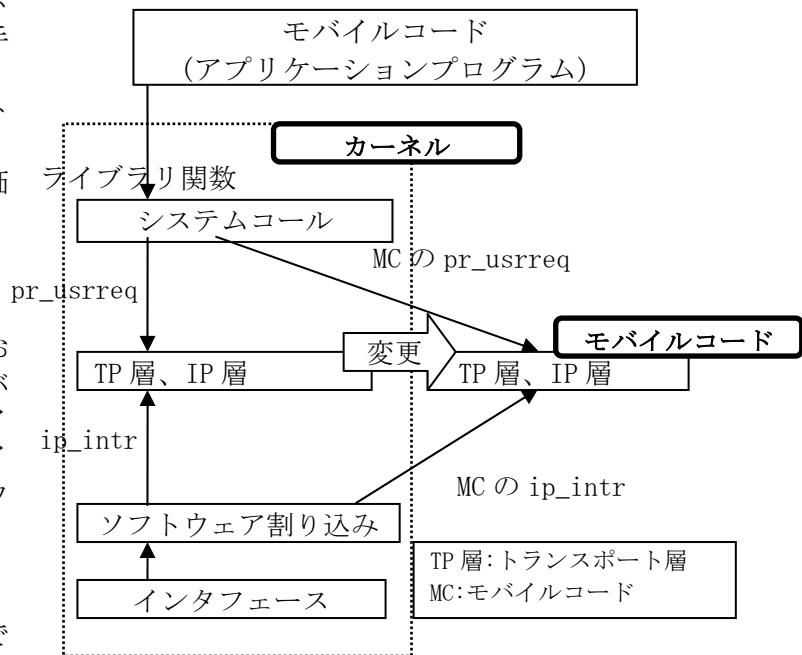


図1 モバイルコードとシステムの関係

トランスポート層のプロトコル機能へのアクセスは、アプリケーションから利用可能なシステムコールsetsockoptで変更できるようにした。IP層への入力ソフトウェア割り込みで呼び出される入力関数ip_intrへのポインタをモバイルコードロード時にモバイルコード内へ変更する。

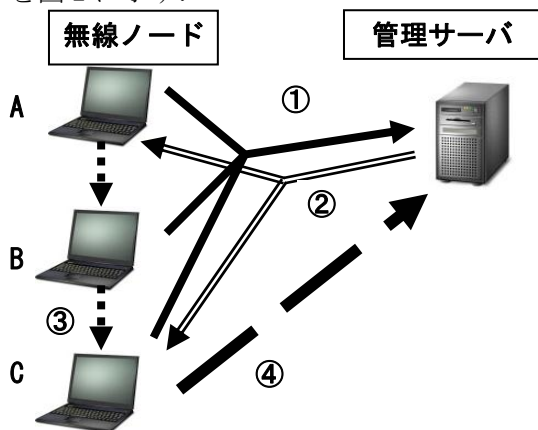
A Development of Wireless Network Simulation Environment using Mobile Code Protocol

Kengo Mikoshiba, Hiroshi Ito, Satoshi Ohzahata, Konosuke Kawashima,

Tokyo University of Agriculture & Technology

2.3 無線シミュレーション管理システム

無線シミュレーション実験を行う際、実験に参加するノードをローカルで操作することなく、管理サーバ上で実験を管理し、実験を自動化することで効率化を図る。実験環境の概要と手順を図2に示す。



- ① IP等の無線ノード情報の収集
- ② 実験測定ツールのモバイルコードおよび、実験シナリオを送信
- ③ 無線ネットワーク実験開始
- ④ 実験結果を管理サーバに送信

図2 無線シミュレーション環境

無線ノード情報の取得には SNMP [1] を使用した。実験ツールの配布、実験内容の指定と配布、および実験の開始は、管理サーバ上の Web 上で設定できる。実験参加ノードをプログラムでリモート管理し、実験終了後に結果を管理サーバの Web 上で閲覧できる。

3. 実装

提案システムを FreeBSD4.7 上に実装した。プロトコルローダは STP [2] を利用するが、アプリケーションプロトコルもロードできるようにローダのシェル部を新たに追加する。モバイルコード化するトランスポート層、IP 層のプロトコルは、トランスポート層は TCP で、モバイルコード内の IP 層にアクセスするようにする。トラフィック生成、測定アプリケーションである ttcp [3] をモバイルコード化して配布する。配布するアドホックネットワークプロトコルは OLSR [4] とし、OLSR が実装されているプログラム OLSRSD [5] を用いた。

4. 実験評価

TCP, IP 部分にモバイルコードを使うようにしたコード内のシステムコールを書き換えた OLSRD (以下、モバイルコード化通信) と、カーネ

ル内の TCP, IP を使う OLSRD (以下、通常通信) の TCP における通信を行った場合の比較を行う。

実験では 3 台の無線ノード (A, B, C) を使用する。A から TCP パケットを送信し、ルーティングプロトコルにより、B を経由し、C でパケットを受信する 2 ホップ通信の配置とした。ノード間の距離は 20m とした。各無線ノードの諸元を次に示す。

PC A 及び B は CPU Pentium II 333MHz, メモリ 64MB. PC C は CPU Celeron M1.3GHz メモリ 256MB. OS は FreeBSD4.7. 無線 NIC は WLI-PCM-L11GP (11Mbps) を全てのノードで使用した。TCP 通信は、データサイズ 1024 バイトのパケットを連続して A から送信し、C で実時間 1 秒あたりのスループットを測定した。

5. 実験結果

表 1 に PC C で測定した通常通信とモバイルコード化通信のスループットを示す。モバイルコード化による著しいネットワークスループットの低下、CPU 使用率の増加は確認できなかったため、提案するモバイルコード化は有効であると考えられる。

表 1 スループット, CPU 使用率

	平均スループット KB/s (標準偏差)	平均 CPU 使用率
通常通信	57.1 (33.72)	0.0027
モバイルコード化通信	56.7 (35.58)	0.0136

6. おわりに

IP, トランスポート層プロトコルをモバイルコード化し、動的にプロトコルを組み込める無線シミュレーション環境を構築した。今後はさらに下位のレイヤーのプロトコルをモバイルコード化する予定である。

参考文献

- [1] J. Case, K. McCloghrie, M. Rose and S. Waldbusser, "SNMPv2," RFC1450, 1993.
- [2] Parveen Patel, Andrew Whitaker, David Wetherall, Jay Lepreau, and Tim Stack, "Upgrading Transport Protocols using Untrusted Mobile Code." Proceedings of the 19th ACM Symposium on Operating System Principles, October 19-22, 2003.
- [3] The Story of the TTCP Program, <http://ftp.arl.mil/~mike/ttcp.html>
- [4] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)," RFC3626, 2003.
- [5] OLSRD, <http://www.olsr.org/>