

# IPSec付き Mobile IPv6 端末エミュレータの試作と評価

2003MT072 中村 光希 2003MT113 山本 美咲  
指導教員 後藤 邦夫

## 1 はじめに

近年、IP 化になった携帯電話や PDA など小型コンピュータの利用が普及してきた。コンピュータの小型化が進むにつれて、それらを持ち運び、場所を問わずに通信する環境が求められつつある。この環境を整えるためには、移動体端末が異なるネットワーク間同士を移動した場合でも、通信が途切れず継続できる新たな機能、つまり IP アドレスの変更なしの通信機能が必要となる。そこで、この問題を解決するために提案されているプロトコル Mobile IPv6 に注目して本研究を進める。

本研究では、ネットワーク部分は既存エミュレータである GINE2R0.5 をルータホストで使用し、複数の MN(Mobile Node) と、HA(Home Agent) を Mobile IPv6 機能を搭載した 1 台の Linux ホストでエミュレーションする。MN ホスト内では、複数の移動先ネットワークを tunnel device である tap デバイスで模倣する。そして、構築したモバイルネットワーク内でテストトラフィックを発生させるため、USB カメラで撮影した映像を配信する。また、多数の MN の移動に伴い、HA にかかる負荷や遅延の検証、映像の品質の性能評価を行う。この際にセキュリティ機能として IPSec を導入し、IPSec 有無の通信速度や遅延の比較も行う。

主に Mobile IPv6 や IPSec の設定、Web サーバ構築、映像アプリケーションの設定を中村が、カーネル構築、HA、端末エミュレータの作成、GINE2R0.5 の設定を山本が行う。

## 2 Mobile IP とは

Mobile IP とは、1996 年に RFC2002 で標準化された技術で、携帯電話などの移動媒体がネットワークを移動することをサポートするプロトコルである。128bit の莫大な IP アドレスを持つ IPv6 の仕組みに対応した Mobile IPv6 を使うことで、電話機に IP アドレスを持たせることや、グローバル IP アドレスを割り振ることが可能となる。Mobile IP では、本来のアドレス CoA(Care of Address) に加え、HoA(Home of Address)、HA を持つことで、移動しながらの通信をサポートする。HoA は、移動端末 MN のホームリンクで持つアドレスを表す。HA は、MN の移動をサポートするホストである。図 1 に移動通信の仕組みを示す [7], [5], [4]。

### 2.1 MN 移動処理

MN は RA(Router Advertisement) 内の prefix 情報の変化によって、ネットワークを移動したことを知る。MN の移動を通知するために、HA に BU(Bindind Up-

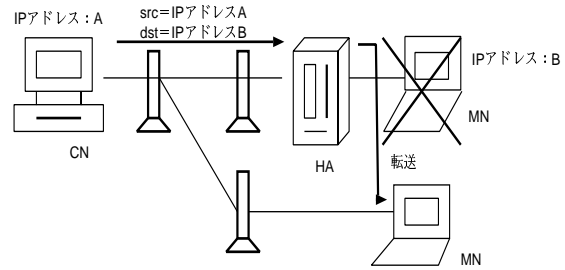


図 1 移動後通信の仕組み

date) を送信する。

- MN はルータから送信される RA を受信して、新しいネットワークに移動したことを知る。また同時に IPv6 のアドレス自動設定の仕組みによって、RA から CoA を生成する
- MN は BU を HA に送信する
- HA は HoA と CoA を管理しているデータベースである Binding cache を更新する

## 3 システムの概要

本章では、本研究のシステムの全体概要と流れについて説明する。

### 3.1 ホスト構成と prefix の設定

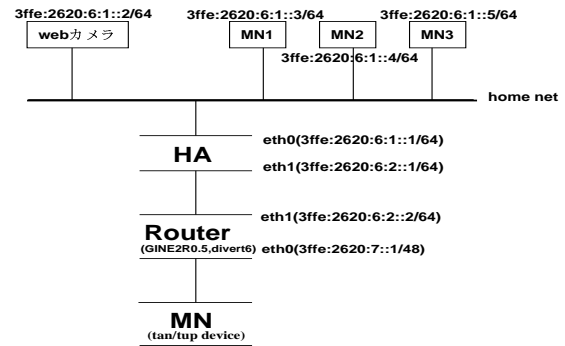


図 2 ネットワーク構成

HA と MN は Mobile IPv6 機能を追加した Linux ホストを各 1 台使用する。また、ルータと USB カメラ用の Web サーバホストには Mobile IPv6 機能がない Linux ホストを各 1 台使用する。ルータは、IPv6DIVER と GINE2R0.5 を入れた障害発生ホストである。Mobile IPv6 は標準でカーネルに組み込まれていないため、Linux カーネルの再構築が必要である。そこで、

HA, MN, ルータに kernel 2.6.15 を, MN と HA に MobileIPv6 機能 mipv6-2.0.1 と, IPSec 機能の ipsec-tools-0.6.2 をインストールした. また, Web サーバは IPv6 で使用可能な Apache 2.2 機能を充てた.

### 3.2 tap インターフェースの利用

1 台の Linux ホストで複数 MN をエミュレーションするには, 以下の 2 つの方法がある.

- パケットキャプチャと RAW ソケットを用いてアプリケーションプログラムで処理する
- Linux tunnel device の活用

前者では, OS の UDP や TCP による処理や, すべてアプリケーションプロセスにより実際の通信が困難である問題点が挙げられる. 逆に, 後者では, OS の中に簡単に搭載できる Mobile IPv6 機能が利用できる利点があるため, 後者を用いることにした. tunnel device である tap デバイスを OS 内で利用するために, 仮想 MN プログラム内で /dev/tun をオープンし, ファイルディスクリプタを利用する. tap デバイスは, パケットを送受信するドライバであり, Ethernet 網のネットワークのインタフェースとしての働きを持ち, 1 度に複数提供できる利点を持つ [3].

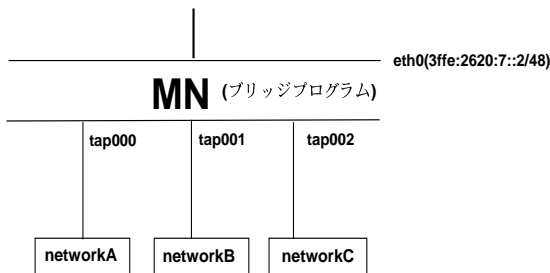


図3 1ホストで複数 MN の実現

### 3.3 IPSec

Mobile IPv6 では, 安全性を保証するために, IPSec というセキュリティ技術を利用している. IPSec とは, IP 暗号技術を用いて, IP パケット単位でデータの改竄防止や秘匿機能を提供するプロトコルである [6]. 本研究では, IPSec によってトンネリングされた両端の HA と MN で手動で IPSec を起動させる. IPSec の設定ファイルでは, アドレス設定や, 暗号 N ベイロード ESP を書き込んだ.

### 3.4 結合更新と擬似 RA

- 結合更新 Binding Update(BU)
 

MN の HoA と CoA の対応関係の情報を含めたものである. HA が BU を受信すると, CN が HoA に送信したパケットを HA が MN の代わりに受信し, CoA に送信することが可能となる [4].
- 擬似 RA 送信 Router Advertisement(RA)
 

ICMPv6 用 raw ソケットで, ネットワークの

prefix 情報や, そのパラメータが表示される [4].

## 3.5 ネットワークエミュレータ

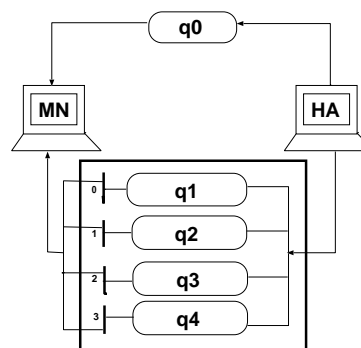


図4 GINE2R0.5 の仕組み

- GINE2R0.5
 

GINE2R0.5 とは, IP diver socket 機能を利用したネットワークエミュレータである. GINE2R0.5 は, 片方向の通信に対して障害を発生させるエミュレータである. 本研究では, HA と MN の双方向の評価を必要とするため, プログラムを変更した. また, MN の移動の都度 CoA が変化するため, prefix アドレスを 4 通りに場合分け, それぞれ異なる障害の発生を可能にした [2].

## 4 端末エミュレータの実現

本研究では, 1 つの MN ホスト内で tap を用いることで, 仮想複数 MN を実現した. 本章では, 端末エミュレータの実現方法について説明する.

### 4.1 複数仮想 MN の実現

1. HA で radvd, Mobile IPv6 を起動
2. 複数 tap インタフェース生成
 

MN で VirtualMN プログラムを実行し tap デバイスを起動, tap インタフェースを生成する. 本研究では, 3 つの tap インタフェースを生成することとする. その後 Mobile IPv6 を起動する.
3. RA 送信
 

tap インタフェース生成後, VirtualMN プログラムが tap インタフェースに RA を送信する. 従って, tap インタフェースにそれぞれのアドレスが自動的に付く
4. BU 送信
 

MN の HoA と tap の CoA 情報を HA に送信する. 従って, HA は初めて MN が異なるネットワークに移動したことを知る
5. HA-MN 間トンネリング
 

移動した MN と HA の仮想ネットワーク間をトンネリング機能を利用して, 通信する

6. MN 移動  
2次元ランダムウォークの計算を元に、MN が他のネットワークへ移動する
7. 3～6 の繰り返し

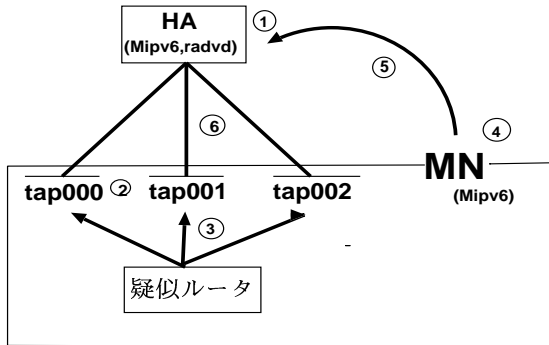


図5 複数仮想 MN の実現

#### 4.2 仮想 MN プログラムの概要

本研究では複数仮想 MN を実現するために、仮想 MN プログラムを作成した。仮想 MN プログラムの内容は以下の通りである。

##### 4.2.1 tap インタフェース

MN 内に tap インタフェースを複数個作成することで、複数 MN を仮想的に実現する。本研究では、3つの tap インタフェース、tap000, tap001, tap002 を作成し、それぞれが MN1, MN2, MN3 の役割を果たす。MN はそれぞれ、仮想 MN プログラムによってネットワークを移動する。また、tap インタフェースを作成し、Mobile IPv6 をそれぞれ起動することで、tunnel device も3つ作成され、トンネリング通信を行う。

##### 4.2.2 擬似 RA 送信の工夫

本来ならば、RA はルータから送信される。しかし、本研究の場合、ルータから RA を送信すると、全ての tap インタフェースに同一のアドレスが付与してしまう。この状況では異なるネットワークを実現できないため、本研究では MN 内で擬似ルータを作成し擬似ルータから擬似 RA を送信する。

本研究では、RA を自分自身に送信することにした。tap インタフェースそれぞれに、別々の RA が送信される。この操作で、tap インタフェースそれぞれに個々のアドレスが付与される。

##### 4.2.3 MN 移動処理

ルータの eth0 のサブネットマスクが 48、MN の CoA のサブネットマスクは 64 なので、先頭から 48 ビット～64 ビット目のアドレス (RA 内の prefix 最後の 16 ビット) の変化で、MN の移動を識別する。このときのアドレス数は  $256 \times 256$  通り存在する。したがって本研究では、x 座標、y 座標を 0～255 用意し、この座標内で 2 次

元ランダムウォークにもとづいて MN の移動を模倣する。移動して導き出された x 座標、y 座標を  $256 \times x + y$  で計算し、出た値を prefix の最後の 16 ビットとする。

## 5 テストトラフィックの発生

構築したモバイルネットワークで性能評価を行うために、テストトラフィックを発生させることにした。

### 5.1 Web カメラ

画像を転送するために、比較的安易に画像を通信できる HTTP を利用することにした。画像は、USB カメラによって撮影する。そして、撮影した画像を一枚毎に連続的に表示することで、動画を模倣することにした [1]。

### 5.2 特定インタフェース (ip6tnl) と bind する専用クライアント

次に画像を受信し、表示するクライアントの設定を行った。本研究では、MN が画像を取得し、インタフェースはトンネルデバイス ip6tnl であり、インタフェースアドレスは HoA である。本研究内では、複数の異なるインタフェースアドレスを持つ MN クライアントを準備している。そのため、画像を表示する MN クライアントを特定する必要がある。一般の Web ブラウザでは、特定のインタフェースを bind できるものはない。したがって、特定の MN クライアントで bind し、画像を取得し、表示することを特定できるプログラムを作成した。

## 6 エミュレータの評価

エミュレータが目的通りの動作をしているか調べた。

### 6.1 複数 MN の移動

移動	tap000	tap001	tap002
1 回目	(45,133)	(176,40)	(101,209)
2 回目	(45,134)	(176,40)	(101,209)
3 回目	(46,135)	(176,41)	(101,208)
4 回目	(46,135)	(176,41)	(101,207)
5 回目	(46,135)	p_off	(100,207)

表1 モバイルノードの移動結果

MN と HA のハードウェアスペックは以下の通りである。

PowerEdge SG430

CPU インテル (R)Pentium(R) CPU 2.8.0GHz

512MB DDR2-SDRAM

クロック周波数 2793.779MHz

この MN で tap を 3 つ起動し、MN が複数ネットワークを移動しているか調べた。表 1 は、MN1 の移動結果である。

### 6.2 GINE2R0.5 を用いた通信性能評価

GINE2R0.5 の遅延とロス値を以下の通り設定し、HA から MN へ ping6 を 100 パケット送り続けた結果、遅延が 500 ミリ秒パケット損失が 7% という結果が得ら

れた。

- queue0=delay(0.1)loss(0.001)
- queue1=delay(0.5)loss(0.0)
- queue3=delay(0.4)loss(0.0)
- queue4=delay(0.3)loss(0.0)
- queue5=delay(0.2)loss(0.0)

loss 値は, ping6 が 64 バイトなので,  $100 * 0.001 * 640 = 6.4\%$ . 上記の値は遅延, ロス値共にほぼ期待値通りと考えられる.

### 6.3 画像品質評価

本研究では, Mobile IPv6 環境を仮想的に実現し, 実際に Web カメラを用いて画像の品質評価を視覚的に行った. 結果は以下の通りである.

- Mobile IPv6 機能だけを用いたとき  
多少の遅れは生じるが, 気にならない
- IPSec を用いたとき  
IPSec を用いてない状態と同じである
- GINE2R0.5 によって遅延, ロスをおこしたとき  
カメラの画像が瞬間的に停止するため, 動画がなめらかでない

動画を見るときに発生する 1MN あたりのトラフィックは, http で送信する jpg 画像 1 枚のバイト数 11000 バイト \* 8 = 88000 ビットとなる. そして,  $88000 * MN$  の専用クライアントで毎秒 get する枚数  $10 = 880000$  より  $880000\text{bps}$  が, 理論上の 1MN あたりのトラフィックとなった.

### 6.4 HA の負荷の検証

仮想 MN の数を 100 個まで増やし, IPSec の有無での HA にかかる負荷の検証を行った. 以下の表 2 は, W コマンドによる負荷平均率である.

MN 数	HA 平均負荷 (1 分) 有	無
10 個	(0.06)	(0.06)
20 個	(0.07)	(0.08)
35 個	(0.08)	(0.14)
50 個	(0.09)	(0.33)
75 個	(1.53)	(-)
100 個	(-)	(-)

表 2 IPSec 有無での HA の負荷

### 6.5 総評

MN の移動先ネットワークを tap を用いて表現することに成功した. MN の移動処理によって HA にかかる負荷を検証するために, tap 数を 100 まで増やし, HA の平均負荷率を検証した. 実験の結果, IPSec 無しの場合, tap 数が 100 で, 有りの場合は tap 数が 75 で, HA の処理が停止し処理不可能となった. この結果, IPSec 機能を加えることで, IPSec 無しするときよりも HA に負荷

がかかり, 通信速度の遅れが発生したことが分かった. 最後に, 構築したエミュレータ内にテストトラフィックを発生させるために, USB カメラによる映像を配信し, 画像の品質を検証した. 画像自体に大幅な遅延は確認できなかったが, HA のハンドオーバー時の BU 処理や, GINE2R0.5 による 100 ミリ秒の遅延により, 画像表示処理が遅れ, 映像に滑らかさがないことが確認できた.

## 7 おわりに

今回の研究では, MN の移動処理を tap を用いて表現し, モバイルネットワークの構築は成功に到った. しかし, tap 数 100 の段階で, HA による処理は限界を示しており, 今後は, この tap エミュレータの性能を高めることが課題の 1 つと言える. また, IPSec 導入や, GINE2R0.5 による通信の遅延や, 動画の品質の違いが, 明確に表れなかった. 今後は, 具体的に通信に影響が表れたときの数値を求め, 様々な状況や遅延率, 負荷率をもとに, 更なる評価を進めることを課題とする. また, 音声などの画像とは異なるアプリケーションプロセスに基づく評価をすることで, 実環境への実装がより早い段階で期待できるであろう.

しかし, Mobile IPv6 は, 実際の環境内で, 普及には至ってはいないため, 利用にはまだ時間がかかることが見込まれる. 実環境ネットワーク内で, Mobile IPv6 機能を用いて, いつでもどこでも監視カメラ映像が見られたり, 自宅内家電の制御が普及するためには, 論述したように, 様々な障害に耐えられるような検証と実験を重ねる必要がある.

## 参考文献

- [1] The Apache Software Foundation: Apache HTTP サーバ バージョン 2.2 ドキュメント (2006). <http://httpd.apache.org/docs/2.2/>.
- [2] Ihara, A., Murase, S. and Goto, K.: IPv4/v6 Network Emulator using Divert Socket, *Proc. of 18th International Conference on Systems Engineering(ICSE2006)*, Coventry,UK, pp. 159–166 (Sep.2006).
- [3] Maxim Krasnyansky: *Universal TUN/TAP driver* (1999-2003). <http://vtun.sourceforge.net/tun/>.
- [4] M.Halpern, J.: 詳細 Mobile IP, プレンティスホール (1986).
- [5] mobile-ipv6.org: *MIPL Mobile IPv6 for Linux* (accessed June 2006). <http://www.mipl.mediapoli.com/>.
- [6] 佐塚秀人: IPsec 入門 (accessed June 2006). <http://cvs.cacanet.org/fsc/ipsec/>.
- [7] 山本孔明: モバイル IPv6 ネットワークのエミュレータの作成とその評価, 卒業論文, 南山大学 数理情報学部 情報通信学科 (2005).