Vol. 41 No. 7 情報処理学会論文誌 July 2000

移動指向ネットワークアーキテクチャの設計と実装

舌間一宏†,☆ 寺岡文男††

ノードの移動は前提であるという観点からネットワークアーキテクチャを再検討し，移動指向ネットワークアーキテクチャを提案する．新アドレス体系として，ノードが接続しているサブネットを示す“ネットワーク指示子”とノードそのものを指し示す“ノード識別子”からなる移動指向アドレスを用いる．トランスポート層より上位層では，ノード識別子のみを用いて通信相手を識別することで移動透過性を実現する．また本アーキテクチャをIPv6に実装し評価を行ったところ，広域ネットワークでの通常の通信時間に比べて，無視できる処理時間で移動透過性が実現できることを示す．

Design and Implementation of Mobility Oriented Network Architecture

Kazuhiro Shitama†,☆ and Fumio Teraoka††

From the viewpoint that every node is mobile, this paper reviews current network architectures and proposes the “mobility oriented network architecture”. The new address format consists of the “network prefix” that specifies the subnet to which the node is connected, and the “node identifier” that identifies the node itself. Making use of the node identifier in the transport and upper layers achieves transparent mobility. The proposed architecture is implemented onto IPv6 and shows that mobility is achieved with negligible processing time compared with communication time in wide area network.

1. はじめに

インターネット上では移動ノードに対応するための様々なプロトコルが数多く提案されてきた2),9)〜12)．現在提案されているプロトコルのほとんどは，ノードの移動を前提としていない既存のインターネットアーキテクチャ上で，移動ノードを特殊なノードとして対応するものである．このアプローチは既存のインターネットとの互換性を考えるうえでは有用であるが，今後モーバイルコンピューティングの形態がさらに多様化し普及していった場合，必ずしも適当な手段とはいえない．そのため，既存との互換性を重視する既存の方式ではなく，ノードの移動は前提であるとしてネットワークアークテクチャ自体を再検討し，既存のインターネットアーキテクチャに代わる，新しいネットワークアーキテクチャの構築が必須であると考えられる13)14)．

本稿では，ノードの移動は前提であるという観点から，モーバイルコンピューティングに適した新たなネットワークアーキテクチャを提案する．特に，通信するノードの位置や移動に関係なく，そのノードと通信できる性質である移動透過性を今後のモーバイルコンピューティングでのキーワードとし，まず，この移動透過性を提供するうえで最適なアドレス体系を提案する．さらに提案したアドレス体系を基にネットワークアーキテクチャの設計を行う．また，提案したネットワークアーキテクチャのプロトタイプを実装し，基本動作に関して評価を行い，本ネットワークアーキテクチャの性能などについて考察する．

2. 移動指向ネットワークアーキテクチャ

2.1 アドレスフォーマット

ノードの移動が前提となる場合に，ノードに対して必要な情報とは以下に示す2点である．

- ノードの移動に対して不変なノード自体を識別するための“ノード識別子”
- ノードまで経路制御を行うためのノードの“位置指示子”

---

† 慶應義塾大学理工学部
Faculty of Science and Technology, Keio University
☆ 現在，ソニー株式会社インフォメーション&ネットワーク研究所
Presently with Information & Network Technologies Laboratories, Sony Corp.
†† 株式会社ソニーコンピュータサイエンス研究所
Sony Computer Science Laboratories, Inc.

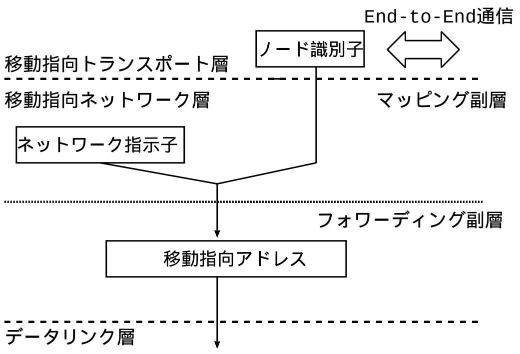

図 1　移動指向ネットワークアーキテクチャ
Fig. 1　Mobility-oriented network architecture.

そこで本稿では位置指示子とノード識別子の 2 つの情報を格納した新しいアドレス体系として，移動指向アドレス（Mobility-oriented Address）を提案する．移動指向アドレスはネットワーク指示子（Network Prefix）とノード識別子（Node Identifier）という，2 つの異なる役割を持つ部分から構成されている．

ネットワーク指示子はノードの位置指示子の役割を果たし，IP アドレスのようにノードのインタフェースの位置を示すのではなく，ノードが接続しているサブネットの位置を示す．ノード識別子は，ノードがどのサブネットに接続しているかに関係なく，ノード自体を識別するための不変的な情報である．ここで示すノード識別子は，MAC アドレスのような NIC（Network Interface Card）を示すインタフェース識別子とは異なるものである．つまり，ノードに挿している NIC を入れ替えることでインタフェース識別子が変わってもノード識別子は不変である．また，ルータのような複数のインタフェース識別子を持つノードでも，そのノードに割り当てられるノード識別子は必ず 1 つとなる．

2.2　プロトコル階層

移動指向ネットワークアーキテクチャは，アプリケーション層，移動指向トランスポート層，移動指向ネットワーク層，データリンク層の 4 つの階層から構成されるネットワークアーキテクチャである（図 1 参照）．

アプリケーション層では，ユーザは通信を行う相手ノードのノード識別子を指定し，各アプリケーションは要求されたサービスをユーザに提供する．ノード識別子を指定することにより，相手ノードの位置を考慮することなく，通信を行うことができる．

移動指向トランスポート層はアプリケーション層に対し，指定されたノード識別子で示される相手ノード上のアプリケーションと自ノード上のアプリケーションにおける End-to-End の通信を提供する．移動に対して不変なノード識別子を用いてコネクションを確立するため，相手ノードや自ノードが移動した場合でも，コネクションは継続される．

移動指向ネットワーク層は上位層で指定されたノード識別子と，そのノード識別子に対応するノードの現在のネットワーク指示子を結合し，移動指向アドレスを生成する．さらに，移動指向アドレスを基に異なるサブネット間の経路制御を行う．データリンク層は既存のインターネットアーキテクチャでのデータリンク層と同じである．

移動指向ネットワークアーキテクチャにおいて 核となる層は移動指向ネットワーク層であり，マッピング副層とフォワーディング副層という 2 つの副層から構成されている．マッピング副層では，上位層で指定されたノード識別子に対応するネットワーク指示子を結合し，移動指向アドレスを生成する．指定されたノードに関するノード識別子とネットワーク指示子の対応づけは，**ACT**（Address Combination Table）で行い，ノードごとの対応づけを **ACT** エントリと呼ぶ．

なお，マッピング副層で相手ノードに関する ACT エントリがない，つまり相手ノードの現在のネットワーク指示子が分からない場合は，相手ノードのホームネットワークを示すネットワーク指示子を相手ノードのノード識別子と結合し移動指向アドレスを生成する．ノードのホームネットワークとは，そのノードが普段接続されているサブネットを示す．ホームネットワークのネットワーク指示子のことをホームネットワーク指示子と呼び，ホームネットワークに接続しているルータをホームルータと呼ぶ．ホームルータは，そのホームルータが管理している移動ノードに関する最新の ACT エントリを保持する．

フォワーディング副層では，マッピング副層で生成された移動指向アドレスを基に経路制御を行い，パケットの送受信および中継を行う．このようにフォワーディング副層は，既存のインターネットアーキテクチャのネットワーク層と同じ機能を持つ．

3. 移動透過性の実現

3.1　ノード間の通信プロトコル

ノード間の通信は以下のように行われる．あるノード CN（Correspondent Node）が，HR（Home Router）をホームルータとする移動ノード MN（Mobile Node）にデータパケットを送信する場合を考える（図 2 参照）．

CN が MN に関する ACT エントリを保持している，すなわち，MN が現在どのサブネットに接続している

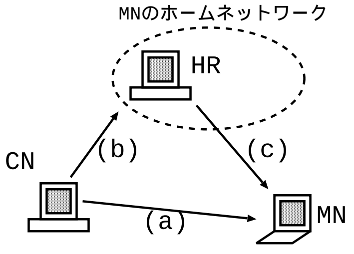

図 2　通信プロトコル
Fig. 2　Communication protocol.

かを CN が知っている場合は，MN が現在接続しているサブネットのネットワーク指示子と MN のノード識別子をマッピング副層で結合して移動指向アドレスを生成する．フォワーディング副層では，その移動指向アドレスに従って経路制御するため，データパケットは，直接 MN に向けて転送される（図 2 (a)）．

一方，CN が MN に関する ACT エントリを保持していない，すなわち，MN が現在どのサブネットに接続しているか知らない場合は，MN のホームネットワーク指示子と MN のノード識別子をマッピング副層で結合して移動指向アドレスを生成する．フォワーディング副層では，その移動指向アドレスに従って経路制御するため，データパケットは，MN のホームネットワークに向けて転送される（図 2 (b)）．そのデータパケットを MN のホームルータである HR が受信すると，HR は MN に関する ACT エントリを必ず保持しているため，受信したデータパケットに記載されている宛先ノードに関するネットワーク指示子部分をホームネットワーク指示子から現在 MN が接続しているサブネットを示すネットワーク指示子に書き換え，MN へ転送する（図 2 (c)）．

3.2　ノードの移動プロトコル

ノードの移動プロトコルを以下に示す．ノード CN と通信中である移動ノード MN が別のサブネットに移動した場合を考える（図 **3** 参照）．なお，MN のホームルータを HR とし，MN が新しく接続したサブネット上のルータを FR（Foreign Router）とする．

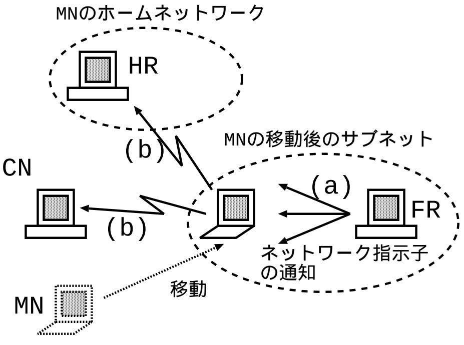

図 3　移動プロトコル
Fig. 3　Node migration protocol.

MN は新たなサブネットに移動すると，接続先のサブネット上の FR から定期的にブロードキャストされているそのサブネットを示すネットワーク指示子を受信する（図 3 (a)）．受信したネットワーク指示子と移動ノードのノード識別子を結合し，移動先での移動指向アドレスを生成する．このように，移動指向アドレスを使用すると，既存のインターネットでの DHCP のようなステートフルアドレス自動設定プロトコル（Statefull Address Autoconfiguration Protocol）は特に必要なく，移動先サブネット上で流れているネットワーク指示子に関する情報を受信するだけで移動先でのアドレスを獲得することが可能となる．

移動先のサブネットで新たに移動指向アドレスを獲得した MN は，HR および CN へ移動通知メッセージを送信し，別のサブネットへ移動したことを報告する（図 3 (b)）．この際に用いられるメッセージを **ACT コントロールメッセージ**と呼ぶ．

なお，ACT コントロールメッセージのやりとりには，ホームルータや通信中の相手ノードに不正な ACT エントリが作成されることを回避するために，メッセージが改竄されていないことと送信ノードの認証を必ず行わなければならない．

ACT コントロールメッセージを受信した CN や HR は，メッセージの送信ノードを認証した後，メッセージに格納されている最新の移動指向アドレスを抽出し，そのノードが保持する ACT エントリに登録する．これにより，MN 宛に送信されたパケットを受信した HR は最新の ACT エントリに従い，MN へその受信パケットを転送することが可能となる．また CN は登録された MN に関する ACT エントリに従って，MN 宛のパケットを HR を経由することなく，直接的に MN へ送信することが可能となる．

なお，自ノードの移動時にのみ ACT コントロールメッセージを送信するだけでなく，自分のホームルータ経由でパケットを受信した場合も，そのパケットの送信ノードが自ノードに関する ACT エントリを保持していないと判断し，その送信ノードへ向けて ACT コントロールメッセージを送信する．

3.3　名前サーバとの連携

ノード識別子はあくまでもビット列であるため，非常に覚えにくく扱いにくい．さらに，2.2 節で述べたように，マッピング副層で通信相手ノードに対する ACT エントリがない場合のために，通信相手ノードのホー

図 4　移動指向 IPv6 アドレスフォーマット
Fig. 4　Mobility-oriented IPv6 address format.

ムネットワーク指示子を獲得するための機構が必要となる．そこで，この 2 点を解決するために名前サーバを利用する．つまり，ノードのホスト名に対してそのノードのノード識別子とホームネットワーク指示子を名前サーバの資源レコードとして格納しておく．

4. 移動指向インターネット MOInet

本稿では，2 章で述べた移動指向ネットワークアーキテクチャのプロトタイプを，IPv6（Internet Protocol version 6）[3]をベースに実装を行った．このプロトタイプを移動指向インターネット **MOInet**（Mobility-Oriented Internet）と呼ぶ．

4.1 移動指向 IPv6 アドレス

2.1 節で述べた移動指向アドレスの概念を適応した IPv6 アドレスを移動指向 **IPv6** アドレス（Mobility-oriented IPv6 Address）と呼び，そのフォーマットを図 4 に示す．移動指向 IPv6 アドレス 128 ビットの上位 64 ビットには，ネットワーク指示子を格納し，下位 64 ビットにノード識別子を格納する．このように 64 ビット + 64 ビットという分割方式を採用した理由は，上位 64 ビットの部分が，IETF で提案されている現在の IPv6 アドレスの仕様[4]での上位 64 ビットと結果として同じ役割となり，既存の経路制御機構を変更しなくてすむからである．つまり，現状のグローバル IPv6 ネットワーク上のルータを変更することなく，本方式を導入することが可能となる．

また，このフォーマットにより，MOInet でサポートできるノード数は 2^{64}（$\cong 10^{19}$）となる．この値は，IETF の IPng Working Group で議論された次世代インターネットとして最低限サポートすべきノード数 10^{12}を十分に満足している[5]．

4.2 MOInet の実装方針

MOInet の実装は，FreeBSD 2.2.7-Release 上で PCMCIA カードをサポートするための PAO パッケージ☆および IPv6 をサポートするための KAME パッケージ☆☆をインストールした PC 上で行った．

2.2 節で述べた移動指向ネットワーク層は，IPv6 層を改良することにより，**moIP** 層として実装を行った．また，トランスポート層では，通信ノードの位置や移動に依存しない End-to-End 通信を提供するために，パケットのデマルチプレックス時は，自ノードおよび相手ノードの移動指向 IPv6 アドレス下位 64 ビットのノード識別子のみに注目し，上位 64 ビットのネットワーク指示子は無視するように改良を行った．

また，3 章で述べたノード間の通信プロトコルやノードの移動プロトコルにおいて必要となる ACT コントロールメッセージの送受信や ACT エントリの追加削除の機能は，ユーザ空間において **actd** デーモンとして実装した．

本実装では，アプリケーションが通信相手ノードのホスト名からノード識別子に変換するときに名前サーバに問い合わせる際に，同時にノードのホームネットワーク指示子を取得するようにした．また，名前サーバにノードのホームネットワーク指示子とノード識別子を格納するための資源レコードを新たに追加するのではなく，すでに用意されている 128 ビットの AAAA レコードの上位 64 ビットをノードのホームネットワーク指示子，下位 64 ビットをノード識別子としたため，名前サーバ自体に特別な改良は行っていない．

ノードの移動検出は，理想的にはハードウェア的に接続リンクの切り替わり（有線リンクならケーブルの抜き挿し，無線リンクなら受信ビーコンの切り替わり）で検出するべきである．しかし，ハードウェア的に検出できなかったり，検出できてもそれをソフトウェア側に通知してくる NIC が現状で手に入らなかったりしたため，本実装では，NIC の抜き挿しによってノードの移動を判断することにした．つまり，ノードから NIC を抜き取ることがサブネットからの切り離しと判断し，ノードに NIC を差し込むことがサブネットへの接続と判断する．そのため，現実装ではユーザはノード移動時にこの作業を明示的に行わなければならないが，NIC の抜き挿しは PAO パッケージに含まれている PC カードの抜き差しの管理を行う pccardd デーモンで自動検出可能であり，抜き挿しが発生すると actd へ移動発生シグナルを発行するように pccardd を一部改良した．移動発生シグナル受信後，actd はルータ申請（Router Soliciation）メッセージを送信し，それを受信したルータが即座にネットワーク指示子を格納したルータ通知（Router Advertise）メッセージをリプライすることで，ネットワーク指示子を取得する[8]．

4.3 ACT の実装

ノードのネットワーク指示子とノード識別子の対応づけを管理する ACT はカーネル内の構造体として実装している．ACT の各エントリのフォーマットを図 **5**

☆ http://www.jp.FreeBSD.org/PAO/
☆☆ http://www.kame.net/

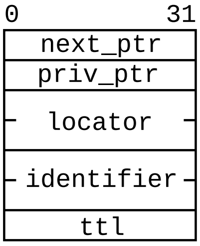

図 5 ACT エントリ
Fig. 5 ACT entry.

に示す．ACT エントリは 5 つのフィールドから構成されており，1 エントリのサイズは 28 バイトである．

ACT エントリはハッシュテーブルにより管理されている．エントリの追加による衝突が発生するとハッシュスロットにリストとして追加される．カーネル内に定義されている ACT の操作関数を以下に示す．

- `moinet_init()` :ACT の初期化
- `moinet_slowtimo()` :エントリのエージング
- `moinet_hashact()` :ハッシュ値の計算
- `moinet_searchact()` :エントリの検索
- `moinet_addact()` :エントリの追加
- `moinet_deleteact()` :エントリの削除
- `moinet_initact()` :全エントリの削除

4.4 moIP 層の実装

moIP 層のブロックダイヤグラムを図 6 に示す．モデル的には，マッピング副層とフォワーディング層を分離して実装するのが理想的であるが，本実装では実装の容易さを考慮して，2 つの副層の明確な分離をしていない．なお，フォワーディング副層の機能は，4.1 節で述べたように既存の経路制御機構を利用できるように移動指向 IPv6 アドレスを定義したため，既存の IPv6 層での`ip6_input()`関数，`ip6_forward()`関数，`ip6_output()`関数の利用が可能である．そこで，マッピング副層の機能を実現するための関数として`moinet_input()`関数，`moinet_output()`関数を新しく作成した．

- `moinet_output()`関数
  宛先アドレスの下位 64 ビットが示す宛先ノード識別子に対応する ACT エントリを検索する．エントリが存在する場合はエントリが示すネットワーク指示子を宛先アドレスの上位 64 ビットとし，ない場合は宛先アドレスの上位 64 ビットをそのまま用いる．
- `moinet_input()`関数
  自ノード宛以外のパケットを受信した場合，宛先アドレス下位 64 ビットのノード識別子に対応する ACT エントリを検索する．エントリが存在する場合は，宛先アドレス上位 64 ビットをエントリで更新する．この動作は，ホームルータのみの機能である．

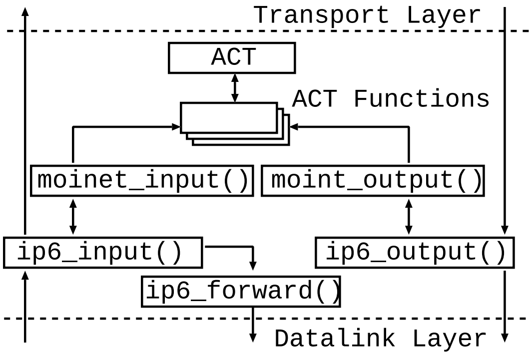

図 6 moIP 層のブロックダイヤグラム
Fig. 6 Block diagram in moIP layer.

4.5 actd の実装

actd は，メッセージ送受信モジュールと ACT 管理モジュールから構成されている．メッセージ送受信モジュールは移動発生シグナルを受信すると，ホームルータおよび通信中の相手ノードへ最新のネットワーク指示子を格納した ACT コントロールメッセージを送信する．また，ACT コントロールメッセージの受信を行い，ACT 管理モジュールに対してエントリ操作要求を行う．ACT 管理モジュールはメッセージ送受信モジュールからのエントリ更新要求にともない，カーネル内の ACT に新規エントリの登録，およびすでに存在するエントリの更新を行う．

メッセージ送受信モジュールでの ACT コントロールメッセージの送受信は UDP を用いて実装している．ACT コントロールメッセージには次に示す 2 つの種類がある．移動通知メッセージは，ノードの移動にともないホームルータおよび通信中の相手ノードに自ノードの最新のネットワーク指示子を通知する場合に用いる．応答メッセージは，受信した移動通知メッセージに対する確認応答として用いられる．なお，3.2 節で述べたように，悪意のあるユーザからの ACT コントロールメッセージによって，ホームルータや通信中の相手ノードに不正な ACT エントリが追加されることを回避するための，移動通知メッセージの正当性と送信ノードの認証には，AH（Authentication Header）6) を用いている．メッセージフォーマットを図 7 に示す．ACT コントロールメッセージは 6 つのフィールド（合計 32 バイト）から構成されている．

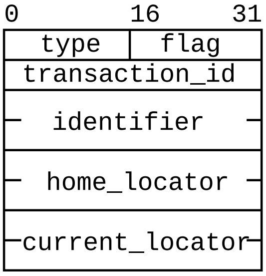

図 7　ACT コントロールメッセージ（UDP）
Fig. 7　ACT control message (UDP).

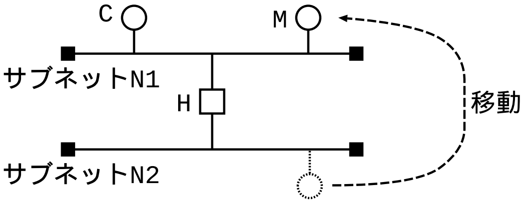

図 8　実験トポロジ
Fig. 8　Evaluation topology.

5. MOInet の評価

5.1 評価項目および環境

図 8 に示す実験ネットワークにおいて，以下の 3 点について評価を行った．

- 通信処理時間：送受信ノードでの moIP 層での処理時間およびホームルータでのパケット転送に要する moIP 層での処理時間
- 登録処理時間：ノード移動時でのコントロールメッセージの送受信および ACT エントリの更新に要する時間
- ACT 操作時間：ACT のエントリ数の増加にともなう，エントリの操作にかかる処理時間

図 8 において，ノード 3 台はすべて同じ性能を持つ PC（CPU Pentium 90MHz，RAM 32M）であり，使用するネットワークは Ethernet（10Mbps）である．ノード M はサブネット N2 をホームネットワークとする移動ノード，ノード H はノード M のホームルータ，ノード C はノード M の通信相手ノードである．

5.2 通信処理時間

ノード C からノード M に向けて ICMPv6 Echo パケットを ping6 コマンドを用いて 100 回送信し，その際の各ノードの moIP 層での処理時間を測定し，その平均値を求めた．なお，ノード C および M には互いのノードに関する ACT エントリではないランダムな 10 個の ACT エントリを，ノード H にはノード M に関する ACT エントリとランダムな ACT エントリ 9 個を，それぞれ追加した状態で測定を行った．ACT のエントリ数が ACT エントリの検索時間にほとんど影響を与えないことは 5.4 節で示す．

表 1　ネットワーク層での基本動作の比較
Table 1　comparisons about basic processings in network layer.

測定項目	moIP 層	IPv6 層	オーバヘッド
送信（ノード C）	19.9 μsec	11.2 μsec	77.7 %
転送（ノード H）	32.1	23.2	38.4
受信（ノード M）	12.0	11.0	9.1

測定結果を表 1 に示す．表 1 で示されるように，送信時および転送時でのオーバヘッドは受信時のそれと比較して非常に大きい．これは，送信時では上位層から指定された宛先ノードに関する ACT エントリの検索，またホームルータでの転送時では受信したパケットの宛先ノードに関する ACT エントリの検索が，それぞれ必要なためである．5.4 節で詳しく述べるように，ACT エントリの検索時間はエントリ数にかかわらず約 7～8μ秒となる．IPv6 層に比べ moIP 層は，送信時で 8.7μ秒，転送時で 8.9μ秒の処理時間が増加していることから，このオーバヘッドの主要因が，ACT エントリの検索時間であることが分かる．それに対して受信時では，moIP 層は IPv6 層と比較して特に特別な処理を行っていないため，処理時間のオーバヘッドは小さい．

送信および転送時での，ACT エントリの検索による処理時間の増加を大きいと見積もるかどうかであるが，この約 10μ秒足らずの処理時間の増加で，既存のインターネットアーキテクチャで不可能であったノードの移動透過性を実現することができるのであれば，十分考慮できるといえる．また，この ACT エントリの検索時間は，通信する 2 ノード間の RTT（Round Trip Time）と比較すれば十分小さく，ほぼ無視できる値といえる．

5.3 登録処理時間

図 8 において，ノード C と通信中のノード M が，サブネット N1 に接続したとする．それにともない，ノード M から ACT コントロールメッセージ（通知メッセージ）を 100 回送信した際の，各ノードでの各処理時間を測定し，その平均値を求めた．ノード C およびノード M の ACT の初期設定は 5.2 節の測定と同様である．

測定結果を図 9 に示す．なお，ノード M でのルータ申請・ルータ通知メッセージの処理時間およびメッセージの RTT は図中の “接続サブネットのネットワー

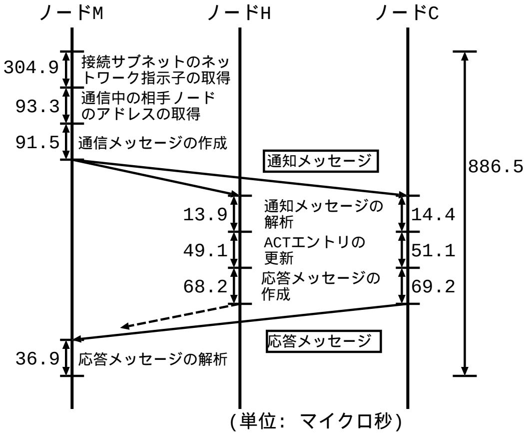

図 9 actd の処理時間の測定結果
Fig. 9 Results of actd processing.

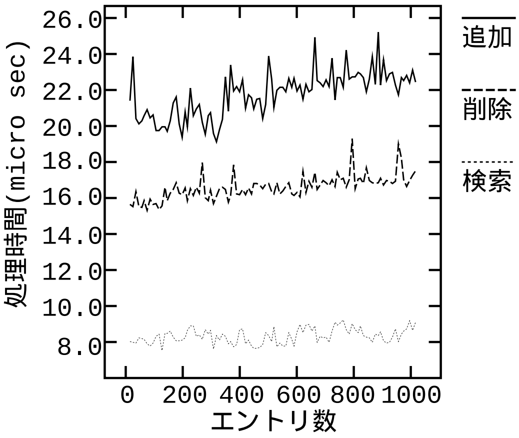

図 10 追加，削除，検索に関する処理時間の測定結果
Fig. 10 Results of add, delete, and search processings.

ク指示子の取得” に含んである．また，ノード M での応答メッセージの受信に関する処理時間は，図の都合上ノード C が送信した応答メッセージに関する処理時間のみを示す．

図 9 において，ノード M で必要な処理時間は約 490μ 秒，またノード C で必要な処理時間は約 130μ秒であり，ノードの移動にともなって各ノードで必要な処理時間の合計は約 620μ秒となる．それに対し，ノード M で移動を検知してから応答メッセージの受信処理が完了するまでにかかる合計時間（移動完了時間と呼ぶ）は約 890μ秒である．

今回の測定は実験環境の制約上，移動ノードと通信相手ノードが同一サブネットに接続している（2 ノード間の RTT が小さい）場合でしか測定できなかったため，各ノードでの処理時間の合計が移動完了時間の約 70%を占めている．しかし，広域ネットワークでのノード間の通信は，通常数ホップ離れた状態で行われる．したがって，通知・応答メッセージの RTT はホップ数に比例して大きくなり，それにともない移動完了時間も大きくなると推測できる．一方，各ノードでの処理時間はノードの位置関係にほとんど依存しない．したがって，広域ネットワークでの通信時間を考えた場合，移動にともなう各ノード内での処理時間の合計は 2 ノード間の RTT と比較してほとんど無視できると考えられる．

5.4 ACT 操作時間

ACT のエントリ数の増加にともなう，検索，追加，削除の処理時間を測定した．図 8 のノード M において，4.3 節で述べた`moinet_searchact()` 関数，`moinet_addact()` 関数，`moinet_deleteact()` 関数を呼び出す`moinet_evalact()` 関数を，測定のためにカーネル内に便宜上作成した．エントリ数が 1，10，50，100，500，1000 の場合に，`moinet_evalact()` 関数が，各関数を呼び出して，その関数から戻ってくるまでの時間を 100 回測定し，その平均値を求めた．ACT 内の各エントリはランダムに作成し，追加するエントリもランダムに作成したものを用いた．検索および削除するエントリは ACT エントリからランダムに抽出したものを用いた．

追加，削除，検索の処理時間の測定結果を図 **10** に示す．図 10 で示されているように，検索，追加，削除とも ACT のエントリ数の増加にかかわらず，多少振動があるがほぼ一定の処理時間となっている．これは，4.3 節で述べたように ACT を線形リストではなく，ハッシュで管理しており，理論上は $O(1)$ のオーダーとなるためである．しかし，エントリ数の増加にともない，ハッシュ値の衝突が発生する確率が高くなるため，どの処理時間も若干ではあるが徐々に線形増加している傾向が読み取れる．

追加および削除の処理時間が，検索の処理時間よりも大きくなっているのは，検索がただ単にハッシュ値の計算および衝突が生じた際のリストの走査だけでよいのに対し，その作業に加えて，追加の場合は新しいエントリのための`MALLOC` のコスト，削除の場合はエントリを削除するための`FREE` のコストがそれぞれ必要となるからである．なお，追加と削除の処理時間の差は，`MALLOC` と`FREE` の処理時間の差と考えられる．

この追加および削除の処理時間が検索の処理時間よりも大きくなることを回避する方法としては，ノードの起動時にあらかじめ決められた数の ACT エントリを`MALLOC` しておき，それらを割当てスプールとして管理する方法がある．追加要求に応じてスプールから

エントリを取り出し，また削除要求に応じてエントリをスプールに戻し，そしてスプールが空になった場合に限り，`MALLOC` を実行するといった作業を行うことにより，追加/削除ごとに毎回`MALLOC`/`FREE` を実行する必要がなくなるため，処理時間が大幅に小さくなると考えられる．

6. 考　察

6.1 Mobile IPv6 との比較

本稿で提案したMOInet は，Mobile IPv6[2),10)]と比較して，TCP/UDP 層でのパケット・デマルチプレックス時にノードの識別をIPv6 アドレス128 ビット全体ではなく下位64 ビットだけで行うように新たに変更する必要があるなど実装上の欠点はあるものの，性能上では以下のような利点を有する．

- IPsec を利用して移動ノードが他のノードと通信している場合，MOInet ではノードが移動してものSA（Security Association）[7)] を維持することができる．Mobile IPv6 ではノードが移動するごとにSA の確立をやり直さなくてはならない．
- MOInet ではIPv6 ヘッダの長さが送信中に変化しない．すなわちIP 層で拡張ヘッダは基本的に付加されない．Mobile IPv6 では，送信ノードのホームアドレスを示す拡張ヘッダがIP 層で付加されるため，アプリケーションやTCP は拡張ヘッダ分をあらかじめ見込んで送信データ長を決めないと，送信する際にフラグメンテーションが発生し，通信効率が低下する．
- MOInet ではノードのネットワークインタフェースが交換されてもノードのアドレスは変化しない．ネットワークインタフェースの交換は，たとえば携帯コンピュータを机上で使用する際には100BASE-TX のPC カードを使用し，会議室に移動する際には無線LAN のPC カードを使用する，などのときに発生する．Mobile IPv6 では，ネットワークインタフェースが交換されるとIPv6 アドレスも変化してしまうため，ホームエージェントへの登録や通信相手とのIPsec のSA の再確立などが必要となる．

6.2 GSE 方式との比較

IETF でのIPv6 の標準化過程において，GSE 方式[1)]と呼ばれるアドレス付け方式が提案された．GSE 方式も128 ビットのIPv6 アドレスを上位64 ビットの経路情報と下位64 ビット位のインタフェース識別子に分離している．本稿で提案した移動指向アドレスとGSE アドレスでは以下の2 点が大きく異なる．

- 移動指向アドレスはノードに割り当てられるのに対し，GSE アドレスはネットワークインタフェースに割り当てられる．したがって，上述したようにネットワークインタフェースの交換があった場合，GSE 方式では基本的にアドレスが変化してしまう．
- GSE 方式では送信ノードは上位64 ビットを空白にして送信し，組織の外に出るときにルータがこの部分に経路情報を代入するが，移動指向アドレスではこのような操作はしない．

また，文献1) は，GSE 方式ではTCP の接続拒否攻撃が容易に実現できるため，セキュリティに問題があると指摘している．これは，悪意のあるノードが別のノードに成りすましてサーバなどにTCP で接続すると，本物のノードから同じサーバへのTCP 接続が拒否されてしまうという問題である．同様の指摘は移動指向アドレス方式にもあてはまる．

しかし，この問題は名前サーバの参照およびホームルータへの転送により容易に回避できる．図**11** のように，悪意のある移動ノードMN' が，HR をホームルータとする移動ノードMN に成りすまして（MN' の送信パケット中の始点アドレスのノード識別子部分にMN のノード識別子を格納して），ノードCN にTCP で接続しようとしたとする（図11 (a)）．MN' からのSYN を受け取ったCN がMN' にSYN+ACK を送信する場合，CN はMN' に関するACT エントリを保持していないため，受信したSYN パケットの始点アドレスのノード識別子部分を名前サーバに問い合わせ，対応するホームネットワーク指示子を得る（図11 (b)）．このホームネットワーク指示子はMN' ではなくMN のホームネットワークを示すため，CN から送信されるSYN+ACK はHR に到着する（図11 (c)）．HR はMN に関するACT エントリを認証による正当な方法で取得しているため，SYN+ACK パケットは悪意のあるノードMN' ではなくMN に到着する（図11 (d)）．したがって，ノードMN' はSYN を送信後，それ以降のTCP パケットを受信することができないため，MN に成りすますことが不可能となる．

6.3 ノード識別子の割当て

64 ビットのノード識別子をどのように割り当て，ユニーク性をどう保証するかについては本アーキクチャを運用していく際，非常に重要な問題である．現在検討中であるが，現時点では次の2 方式を考えている．

- IP アドレス割当てのように，ARIN（American Registory for Numbers），RIPE（Reseaux IP Europeens），APNIC（Asia Pacific Network In-

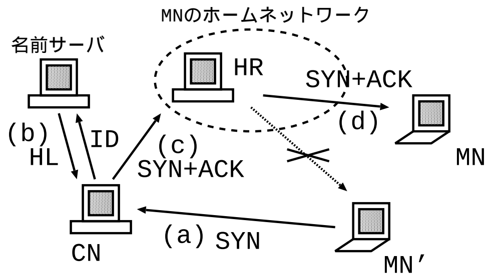

図 11　TCP の接続拒否攻撃に対する解決方法
Fig. 11　Solution to TCP service denial attack.

formation Centre）などのレジストリ機関から下部組織に階層的にノード識別子を割り当てる方式

- 今後，インターネットに接続する可能性のある機器には，ノード識別子として EUI-64（ユニーク性を保証するために IEEE が規定する 64 ビットの番号）を割り当てて出荷する方式

7. おわりに

本稿ではノードの移動は前提であるという観点から移動指向ネットワークアーキテクチャを提案した．さらに，本アーキテクチャを IPv6 に適用した MOInet を実装・評価したところ，広域ネットワークでの通常の通信時間に比べて，無視できる処理時間で移動透過性が実現できることを示した．

参考文献

1) Crawford, M., Mankin, A., Narten, T., Stewart III, J.W. and Zhang, L.: Separating Identifiers and Locators in Addresses: An Analysis of the GSE Proposal for IPv6, Internet-Draft, work in progress, IETF (1999). ftp://ftp.ietf.org/internet-drafts/draft-ietf-ipngwg-esd-analysis-05.txt.

2) David, J. and Perkins, C.: Mobility Support in IPv6, Internet-Draft, work in progress, IETF (2000). ftp://ftp.ietf.org/internet-drafts/draft-ietf-mobileip-ipv6-10.txt.

3) Deering, S. and Hinden, R.: Internet protocol, version 6 (IPv6) specification, RFC 2460, IETF (1998).

4) Hinden, R., O'Dell, M. and Deering, S.: An IPv6 Aggregatable Global Unicast Address Format, RFC 2374, IETF (1998).

5) Kastenholz, F. and Partridge, C.: Technical Criteria for Choosing IP: The Next Generation (IPng), RFC 1726, IETF (1994).

6) Kent, S. and Atkinson, R.: IP Authentication Header, RFC 2402, IETF (1998).

7) Kent, S. and Atkinson, R.: Security Architecture for the Internet Protocol, RFC 2401, IETF (1998).

8) Narten, T., Nordmark, E. and Simpson, W.: Neighbor Discovery for IP Version 6 (IPv6), RFC 2461, IETF (1998).

9) Perkins, C.: IP Mobility Support, RFC 2002, IETF (1996).

10) Perkins, C. and David, J.: Mobility Support in IPv6, *Proc. MOBICOM '96* (1996).

11) Teraoka, F. and Uehara, K.: Mobility Support in IPv6 based on the VIP mechanism, *Proc. INET'95* (1995).

12) Teraoka, F., Uehara, K., Sunahara, H., and Murai, J.: VIP: A Protocol Providing Host Mobility, *Comm. of the ACM*, Vol.37, No. 8, pp. 67–75 (1994).

13) 舌間一宏，寺岡文男：IPv6 における移動透過性の実現，情報処理学会モーバイルコンピューティング研究会第 5 回研究会報告集 (1998).

14) 舌間一宏，寺岡文男：v6VIP の設計と実装，情報処理学会マルチメディア，分散，協調とモーバイル（DICOMO'98）シンポジウム論文集 (1998).

(平成 11 年 9 月 16 日受付)
(平成 12 年 5 月 11 日採録)

舌間 一宏

1974 年生．1997 年慶應義塾大学理工学部電気工学科卒業．1999 年同大学大学院理工学研究科計算機科学専攻修士課程修了．同年ソニー（株）入社．ネットワークアーキテクチャ，モーバイルコンピューティングに興味を持つ．

寺岡 文男（正会員）

1959 年生．1984 年慶應義塾大学大学院修士課程修了．同年キヤノン（株）入社．1988 年（株）ソニーコンピュータサイエンス研究所入社．工学博士．1991 年日本ソフトウェア科学会高橋奨励賞受賞．1993 年元岡記念賞受賞．コンピュータネットワーク，オペレーティングシステム，分散システム等の研究に従事．特に移動透過性を提供するプロトコル VIP（Virtual IP）の開発を通して IETF の Mobile IP 分科会の活動に貢献．2000 年 5 月から情報処理学会理事．ACM，IEEE，日本ソフトウェア科学会各会員．