

IPv4 拡張した移動透過通信アーキテクチャ MAT の設計と性能評価

関 顕 生^{†1} 岩 田 裕 貴^{†2} 森 廣 勇 人^{†2}
前 田 香 織^{†2} 近 堂 徹^{†3} 岸 場 清 悟^{†3}
西 村 浩 二^{†3} 相 原 玲 二^{†3}

近年、ラップトップ PC などの移動端末の普及や無線 LAN 環境の整備により、移動通信への期待が高まりつつある。筆者らは、IP 層において移動透過通信を実現するアーキテクチャ MAT を提唱し、IPv6 での利用を想定して具体的なプロトコルの設計、実装を行ってきた。しかし、移動端末を IPv6 で利用できる環境は少なく、ISP が提供するサービスは現在でも IPv4 が中心であるため、MAT を IPv4 にも対応するよう拡張し、具体的なプロトコルの設計を行った。本稿では、MAT の IPv4 への拡張点について述べ、移動ノードの通信開始動作とハンドオーバー動作、IPv4 でのシグナリング手順などについて述べる。また、Linux 実装について述べ、ハンドオーバーを含む通信性能の測定結果を示す。

Design and Evaluation of IP Mobility Architecture MAT Extended to IPv4

AKIO SEKI,^{†1} YUKI IWATA,^{†2} HAYATO MORIHIRO,^{†2}
KAORI MAEDA,^{†2} TOHRU KONDO,^{†3} SEIGO KISHIBA,^{†3}
KOUJI NISHIMURA^{†3} and REIJI AIBARA^{†3}

Recently, mobility support communications are required because laptop PCs, portable devices and wireless LAN and/or WAN infrastructures have been widely deployed. We have developed an IP mobility architecture MAT for IPv6 which enables the mobile node to communicate transparently even if it moves from one network to another. However, wireless LAN/WAN environments for portable devices are currently restricted. The purpose of our study is to apply the technology of MAT which has been cultivated in IPv6 to existing IPv4 networks. In this paper, we explain protocol extensions of MAT for IPv4, and how signaling messages are exchanged among nodes and servers. Finally,

we describe about Linux implementation of our proposal, and demonstrate the results of evaluation including handover phase.

1. はじめに

近年、移動端末の普及や無線ネットワーク環境の整備により、いつでもどこからでもインターネットへの接続が可能になりつつある。このような状況下では、通信中にユーザが移動しても、通信を継続できる移動透過通信機能が求められる。インターネットでは、IP アドレスがノードの識別子としての役割だけでなく、IP アドレスのネットワーク部分が間接的に位置情報も示しているため、端末が別のネットワークへ移動すると異なる IP アドレスが割り振られる。トランスポート層では IP アドレスが通信識別子の一部に用いられているため、IP アドレスの変更により別の通信と見なされ通信を維持することができない。このような背景から、IP 層における移動透過通信アーキテクチャの研究が行われている。IPv4 で移動透過通信を実現する技術として、Mobile IP¹⁾、Mobile PPC²⁾、IPv6 では Mobile IPv6³⁾、LIN6⁴⁾、MAT⁵⁾ などが提案されている。

筆者らは IPv6 ネットワークでの利用を想定して MAT を提案し、実装を行ってきた。しかし、IPv6 ネットワークは着実に普及しつつあるものの、ISP が提供する無線系サービスは今でも IPv4 が中心である。IPv4 では Mobile IP が提案されているが、このアーキテクチャは通信ノード間が非最適経路となる性質を持つ。今日ではノート PC 以外にもスマートフォンなどの IP 対応モバイル機器が爆発的に増加しており、モバイル通信のトラフィックがインターネットに与える影響を考えると、つねに最適経路で通信する Mobile PPC や MAT を使用することが望ましい。なお、MAT は Mobile PPC が対応していないコネクションレス型の通信も移動透過通信の対象としている。また、IPv4 から IPv6 へのネットワーク移行期間中には、両方のプロトコルで移動透過通信が可能であることが強く望まれる。そこで、MAT を IPv4 ネットワークにも対応できるように拡張し、IPv4 と IPv6 の両方に対応した移動透過通信基盤を同一のリソース (IMS などの通信継続性のための管理資

†1 広島大学大学院総合科学研究科
Graduate School of Integrated Arts and Sciences, Hiroshima University

†2 広島市立大学大学院情報科学研究科
Graduate School of Information Sciences, Hiroshima City University

†3 広島大学情報メディア教育研究センター
Information Media Center, Hiroshima University

源)で対応できるようにした。これにより、移行期間中でもIPモビリティのための設備投資を最小限に抑えることができる。実在のIPv4ネットワークでは、拡張したIPオプション付きデータグラムを多くのルータが破棄する設定となっているため、MAT4の Protokol設計ではIPオプションを使用しないこととした。その結果、IPv6の場合より制御メッセージ数が増加するため、その影響を詳細に評価する必要がある。以降、MATの実装を区別するため、既存のMATをMAT6、IPv4対応のMATをMAT4と記す。

本稿では、まず2章で関連研究の紹介を行い、3章でMAT4の構成とProtokolの設計について述べ、4章でMAT4ノードの制御方法について述べる。5章でMAT4のLinux実装の性能評価を示す。最後に6章でまとめを述べる。

2. IPv4 移動透過通信に関する関連研究

移動透過通信の研究は先述したとおり、様々な方式が提案・実装されており、それらはプロキシ方式とエンドツーエンド方式に大別することができる。ここではプロキシ方式の例としてMobile IPを、エンドツーエンド方式の例としてMobile PPCとMAT6を取り上げる。

2.1 Mobile IP

Mobile IPはIETFにおける議論を経て標準化されたIPv4用の通信技術である。移動ノード(MN: Mobile Node)はノード識別子としてホームアドレス(HoA: Home Address)と、位置識別子として移動先のネットワークから割り当てられる気付アドレス(CoA: Care-of Address)の2つのIPアドレスを利用する。ホームエージェント(HA)は、HoAとCoAの対応付け(バインディング)を行い、CoAを持つMNへ転送する役割を持つ。

MNは異なるネットワークへ移動したとき、移動先のネットワークで新しいCoAをHAに登録し、HAはMNのバインディングを更新する。通信相手ノード(CN: Correspondent Node)からMNへパケットを送信する場合は、宛先アドレスをHoAとしHAへ送信する。CNからのパケットを代理受信したHAは、パケットをMNのCoAでカプセル化してMNへ転送する。一方、MNからCNへのパケットは送信元アドレスをHoAとし、HAを経由せずに直接送信される。

Mobile IPは、HAというプロキシサーバを導入し、CNがつねにHAと通信しているように見せかけることにより移動透過性を獲得している。Mobile IPには、(1)通信経路が冗長な三角経路になること、(2)HAが単一障害点となること、(3)MNからCNへのパケットは送信元アドレスにHoAを使用しており、位置識別子としてMNのネットワーク上の位

置を正しく表していないこと、などの問題点⁶⁾が指摘されている。

2.2 Mobile PPC

Mobile PPCはエンドツーエンドでアドレス変換処理を行う移動透過通信技術である。この手法はエンドツーエンド方式であるため最適経路で通信ができ、IPトンネルが存在しないので高スループットを実現できる。また、通信の継続に必要な情報をMNとCNが直接通知するという特徴があり、Mobile IPの単一障害性を克服している。Mobile PPCの動作は、移動情報の通知とIPアドレスの変換によって機能する。MNが別のネットワークへ移動した場合、移動先のネットワークで取得した新しいIPアドレスを直接CNへ通知する。この通知処理により、MNとCNは各ノード内で保持している移動前と移動後のIPアドレスの対応関係を示すCIT(Connection ID Table)を更新する。CITレコードは通信開始時にコネクション単位で作成されるものであり、MNが移動するたびにその内容が更新される。コネクションごとに通信開始時のIPアドレスと移動後のIPアドレスとの変換が行われるため、上位層には移動によるIPアドレスの変化が隠蔽され、通信を維持することができる。

Mobile PPCは、MNが移動先で得た新しいIPアドレスを直接CNへ通知する方式であり、コネクションの略奪を防ぐため、通信開始時にノード間の認証を必要とする。そのため、通信開始時の鍵交換などのためのオーバーヘッドが大きくなる。さらに、UDPなどのコネクションレス通信に対する移動透過性を提供できない。

2.3 MAT6

MAT6は筆者らが提案し実装を行ってきたエンドツーエンド方式による移動透過通信アーキテクチャである。MAT6はIPアドレスが持つノード識別子とネットワーク上の位置識別子の性質を2つのアドレスに分離し、前者をホームアドレス(HoA: Home Address)、後者をモバイルアドレス(MoA: Mobile Address)と呼ぶ。HoAはMNのアプリケーションが使用するIPアドレスであり、MoAはネットワークから一時的に割り当てられるものである。HoAとMoAの対応関係をマッピング情報と呼び、IMS(IP address Mapping Server)で管理している。IMSは耐障害性の観点から冗長構成をとることができる⁷⁾。MNとCNは、マッピング情報に基づいて各ノードのIP層でアドレス変換することにより、アプリケーションに対して移動にともなうIPアドレスの変更を隠蔽できるため、継続的な通信が可能となる。

MAT6で使用するDNSは、各ノードのマッピング情報を管理するIMSのIPアドレスを提供できるように通常のDNSから拡張が行われている。ただし、この拡張はMNやCN

の HoA に属するゾーンを委譲されたサーバにのみに施されていけばよい。MN と CN がいずれも MAT に対応しなければならないという制約があるものの、MAT は前述の Mobile IP の問題点および Mobile PPC がかかえる制限事項を解決する。

MAT は IPv6 での研究が先行しており、実装も IPv6 を対象に行ってきた。MAT6 の概念そのものは IPv4 に適用することが可能であるが、仕様変更も必要である。MAT6 では制御メッセージの一部に IPv6 拡張ヘッダを使用している。IPv4 にも IP オプションが定義されているが、実際のネットワーク環境ではセキュリティ対策により、未定義 IP オプションの付いたパケットは破棄する設定となっている場合が多い。そのため、IP オプションを使用しない制御機構を設計する必要がある。

3. MAT4

3.1 MAT4 の構成

図 1 に MAT4 の構成を示す。本節以降では、CN は MAT4 対応ノードとする。MN は MAT の実装を持たないノードとも通信することが可能であるが、その場合移動透過通信を行うことはできない。図 1 中の IMS_{MN} は MN 用の IMS を、 IMS_{CN} は CN の IMS を表す。実装はこれらを 1 台の IMS で運用することが可能であるが、IMS を分散設置することも考慮し、プロトコルとしてはそれぞれを区別して設計した。Mobile IP では通信の継続に必要な HA が単一障害点となるが、MAT は MN の HoA を管理する IMS を冗長構成とすることで耐障害性を高めることができる⁷⁾。

MAT4 と MAT6 は、互いに独立した IP 層プロトコルであり、同一ノードにデュアルス

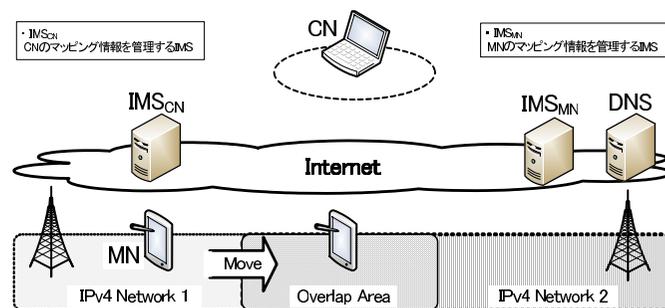


図 1 MAT4 の構成

Fig. 1 Structure of MAT4 systems.

タック構成で実現することが可能である。移動透過通信もそれぞれ独立に動作することとし、IPv4 と IPv6 間でのハンドオーバーは考慮していない。IMS もデュアルスタック構成であり、同じリソースで MAT4 と MAT6 に対応することができる。そのため、IPv4 から IPv6 への移行期間中であっても、MAT4 と MAT6 で同じ IMS を使用でき、IMS の特別な運用管理は不要である。

3.2 プロトコルの設計

MAT4 のプロトコル設計要件を以下にあげる。

- (1) MAT4 ノードの制御メッセージとして IP オプションを使用しない。
- (2) CN は MN のマッピング情報を IMS_{CN} から取得する。
- (3) IMS_{MN} - IMS_{CN} 間でマッピング情報の受渡しを行う。

MAT6 では、ハンドオーバー時に自分のモバイルアドレス(マッピング情報)が変化したことを CN に伝えるために、IPv6 拡張ヘッダ (MUO: Mapping Update Option) を定義して利用している。IPv4 にも IP オプションは利用可能であるが、IPv4 環境における未定義 IP オプションの付いたパケットは、セキュリティ対策などによりネットワーク経路上で破棄されることがしばしばあるため、使えない場合が多い。(2) と (3) は、MAT6 において MN が MUO を用いてマッピング情報を CN に直接通知していたことを、MN が IMS_{CN} 経由で CN に通知するために追加したプロトコルである。

3.3 MAT4 が動作するネットワーク環境

MAT4 では、MN と CN が使用する MoA はグローバルアドレスであり、MN, CN, IMS 間の通信経路上に NAT などが入っていないことを前提としている。これは、MAT4 におけるマッピング情報はノード識別の HoA とグローバル到達性を持つ MoA との対である必要があり、通信経路上に NAT が存在すると相手ノードへの到達性が確保できないためである。さらに、4.1 節で述べる MAT4 のシグナリングのための通信も許可されている必要がある。

4. MAT ノードの制御

4.1 MAT4 のシグナリング

MAT4 のシグナリングメッセージは、アドレス変換処理に必要なマッピング情報を MN-IMS-CN 間で相互に更新・通知・参照するために用いるものである。メッセージは要求メッセージと応答メッセージの 1 組で機能する。

MAT4 では、次の 4 メッセージを新しく追加した。

- HA 通知 (HA Notify)

MN が IMS_{CN} へ MN の HoA を通知する .

- HA 通知応答 (HA Notify Ack)

HA 通知の応答を IMS_{CN} が MN へ通知する .

- HA 問合せ (HA Query)

MAT 対応ノードが自身を管理する IMS へ通信相手の MoA をキーにして HoA を問い合わせる .

- HA 応答 (HA Answer)

HA 問合せの回答を IMS が MAT 対応ノードへ通知する .

また、次の 2 メッセージは MAT6 において (MN,CN)-IMS 間で使用しているが、MAT4 ではさらに IMS-IMS 間でも使用する .

- MA 問合せ (MA Query)

MAT 対応ノードが通信相手の HoA をキーにして、IMS に MoA または Mobile Prefix (MoPre) を問い合わせる . また MAT4 では、 IMS_{CN} が MN の HoA をキーにして IMS_{MN} に MoA を問い合わせる .

- MA 応答 (MA Answer)

IMS が MA 問合せの回答を、MAT 対応ノードあるいは要求のあった IMS へ通知する . 次の 2 メッセージは MAT6 と MAT4 の両方で使用する .

- IMS 更新要求 (IMS Update)

MAT 対応ノードが自身のマッピング情報の更新を自身を管理する IMS に要求する .

- IMS 更新通知 (IMS Update Reply)

IMS 更新の成否を IMS が MAT 対応ノードへ通知する .

今回の実装では、上記のシグナリングはすべて UDP を使用しており、MN、CN、IMS 間に NAT が 1 つ以上存在している場合、たとえば、通知メッセージが到達してもそれに対する応答メッセージが正常に返らない場合がある . シグナリングを TCP に変更することで、一定条件を満たす NAT を経由させることは可能となるが、3.3 節で述べたとおり、MAT4 はグローバルアドレス、グローバル到達性を前提としており、NAT などを経由しないことを動作条件としている .

IPv4 環境では、グローバルアドレスを使用し、グローバル到達性がある場合であっても Firewall などによって特定の packets を破棄している場合がある . MAT4 を使用するには、MAT4 のシグナリング通信が破棄されないことが必要条件となる . 今回の実験では、IMS の受信に UDP 10121 番ポートを、MN および CN の受信に UDP 40930 ~ 40940 番

ポートを使用したがる、ポート番号は設定変更可能である .

なお、MN と CN 間の通信そのものが、途中で存在する Firewall などによって破棄対象となっている場合は、いうまでもなく目的の通信を行うことができない .

4.2 通信開始手順

図 2 の (1) から (7) に MAT4 の通信開始手順を示す . 図 2 の実線はシグナリングメッセージの流れを、破線はユーザデータの流れを示している . ただし、DNS との通信は省略している . MAT6 と同様に、MN は CN との通信開始時に、 IMS_{CN} の IP アドレスを取得する . (1) MN は IMS_{CN} に MA Query を送り、その応答により CN のマッピング情報を取得する . (2) 次に MAT6 の MUO に相当する HA Notify を IMS_{CN} に送信する . このメッセージを受信した IMS_{CN} は IMS_{MN} の IP アドレスを DNS により取得し、(3) IMS_{MN} に MA Query を送り、その応答により MN のマッピング情報を取得する . (4) MN は、 IMS_{CN} からの HA Notify Ack によって通信準備完了を確認し、(5) CN との通信を開始する . MAT4 では (1) から (4) のステップを踏むことで、CN が IMS_{CN} から MN のマッピング情報を取得できるようにしている . なお、 IMS_{MN} と IMS_{CN} が同一ホストである場合は、(3) のシグナリングが省略される .

CN が MN から初めて通信パケットを受信したとき、MN のマッピング情報を知らないためアドレス変換を行うことができない . そのため、(6) CN は知らない MoA を送信元と

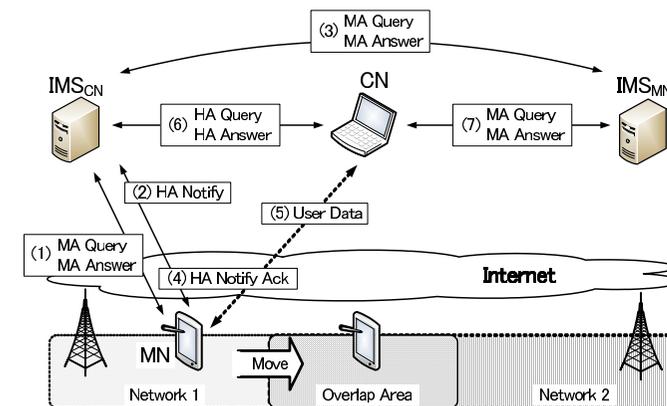


図 2 MAT4 の通信開始手順

Fig. 2 Initial communication procedure in MAT4.

する通信パケットを受信すると、その MoA をキーに IMS_{CN} へ HA Query を送り、その応答で対応する HoA を取得する。(4)の時点で IMS_{CN} 上には MN のマッピング情報が存在しているので、CN に MN の HoA を提供することが可能となっている。MN の HoA を取得した CN はこの時点で MN と相互通信ができるようになる。さらに CN は IMS_{MN} の IP アドレスを DNS により取得し、(7) IMS_{MN} に MA Query を送り、その応答で MN のマッピング情報を取得する。CN は (6)の時点で MN のアドレス変換に必要なマッピング情報を揃えることができるが、MAT6 と同様に、MN 自身によってマッピング情報が更新された IMS を信用する。以降、通信中の MN と CN は、通信相手の IMS に定期的に MA Query を行い、互いに最新のマッピング情報の取得を行う。

4.3 ハンドオーバー手順

図 3 に MAT4 のハンドオーバー手順を示す。MAT4 は、MAT6 と同様に複数インタフェースの使用を推奨している。MN が移動先である Network 2 に対しても接続ができるようになり、通信に使用していないインタフェースに DHCP⁸⁾ によって MoA が割り当てられると、(1) IMS_{MN} に IMS Update を送りマッピング情報を更新する。その応答が返ってくると MN は、(2) HA Notify を IMS_{CN} へ送り、自身のマッピング情報が更新されたことを通知する。この通知を受信した IMS_{CN} は、(3) IMS_{MN} へ MA Query を送信し、その応答により MN のマッピング情報を取得し、 IMS_{CN} 上にある MN のマッピング情報

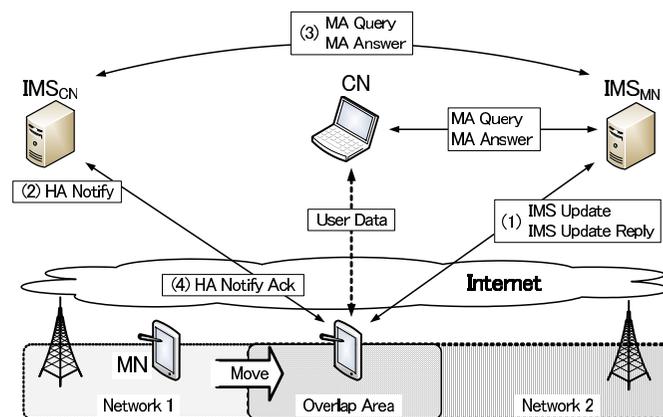


図 3 MAT4 のハンドオーバー手順

Fig. 3 Handover procedure in MAT4.

を更新する。(4) IMS_{CN} は、MN に通知完了確認となる HA Notify Ack を送る。なお、 IMS_{MN} と IMS_{CN} が同一ホストである場合は、(3)のシグナリングが省略される。

CN は IMS_{MN} への定期的な MA Query で、(1)の後の MN の最新のマッピング情報を取得する。MN が Network 2 に接続したインタフェースを用いて通信すると判断して通信を切り替えても、CN が保持するマッピング情報には Network 2 上の MN の MoA が含まれているため、即座にデータを受け取ることができる。

しかし、CN が最新のマッピング情報を受け取る前に、MN が Network 2 の MoA を用いて通信してきた場合は、すぐにアドレス変換を行うことはできない。この場合は、図 2 の (6)と (7)の手順が発生することになる。このような状況は、MN が移動するネットワーク間に十分なオーバーラップエリアが存在しないときに起こりうる。

また、MN がシングルインタフェースでハンドオーバーを行うときは、少なくとも MN が Network 1 から切断し、Network 2 において (4)の HA Notify Ack を完了するまでの間、CN と通信不能な時間が発生する。

5. MAT4 の性能評価

この章では MAT4 の Linux 実装について IMS と MN の性能評価を述べる。移動透過通信アーキテクチャでは通信継続性を提供するサーバの性能がモバイルノードの通信に影響を与えるため、5.1 節で IMS における MAT6 と MAT4 の処理性能を比較する。5.2 節では、(1)スループット性能と(2)リンクアップから通信可能になるまでの時間、(3)ハンドオーバー時のパケット到達遅延時間について実験内容と測定結果を述べ、具体的な通信性能の目安を示す。また、5.3 節でこれらの実験結果の考察を述べる。

5.1 IMS

IMS は、HoA と MoA の対応関係であるマッピング情報を提供する。MAT4 に対応するために、従来の IMS を拡張し、MAT4 ノードと IMS 間のシグナリングサポートを追加した。拡張点は、以下の 3 点である。

- (1) IPv4 と IPv6 の両プロトコル対応
- (2) HA Notify/Ack とそれにもなう MA Query 処理
- (3) HA Query/Answer 処理

実験では、電車やバス内で移動端末を持っている人がネットワークのオーバーラップが狭いエリアでいっせいにハンドオーバーするという状況を想定し、IMS におけるハンドオーバーの処理時間を測定、MAT6 と MAT4 で動作特性を比較した。

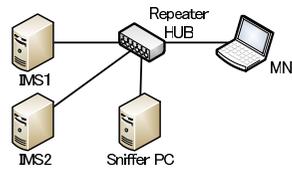


図 4 IMS 性能測定構成
Fig. 4 Measurement system of IMS.

表 1 IMS 測定機器の諸元
Table 1 Node specifications in IMS measurement.

Node	Hardware	OS
IMS1, IMS2	Intel Core2 Quad 2.3 GHz 4 GB RAM	Debian 5.0 linux-2.6.26
MN	Intel Core2 Duo 2.2 GHz 2 GB RAM	Debian 5.0 linux-2.6.26
Sniffer PC	Intel Xeon 3.8 GHz 2 GB RAM	Windows XP SP3

表 2 IMS のアドレス登録数

Table 2 Number of address entries in IMS.

IPv6 address		IPv4 address		Total entry
HoA	MoA	HoA	MoA	
1,000,000	2,000,000	65,534	131,068	3,196,602

図 4 に測定構成を示し、表 1 に測定に使用した機器の諸元を示す。2 台の IMS のうち、処理時間の測定対象となるのは IMS1 であり、IMS2 は MAT4 用に IMS1 からの MA Query を処理するために用意している。各 IMS には表 2 に示すアドレス数を登録しており、図 4 の MN が一定の頻度でハンドオーバー動作を 100 ミリ秒間実行した場合の平均処理時間を求めた。ハンドオーバーの所要時間は、MAT6 は MN の IMS Update と CN の MA Query の合計時間、MAT4 は MN の IMS Update と HA Notify、CN の HA Query と MA Query の合計時間とし、毎秒 10 から 20,000 までのハンドオーバー頻度で測定した。各測定におけるメッセージの送信時間は等間隔とした。

5 回の測定における平均値を図 5 と図 6 に示す。MAT6 は、毎秒 20,000 のハンドオーバーにおいても所要時間は約 0.4 ミリ秒であった。一方で MAT4 は、処理数が毎秒 8,000 までは約 1.4 ミリ秒であり、8,000 を超えると処理時間が大幅に増加する結果となった。これは MAT4 の HA Notify が影響している。IMS1 では HA Notify に対応する MA Query の処理が発生し、この測定で使用した IMS1 のリソースに対して処理数 8,000 を超えると処理負荷が高くなるために処理時間が増加したと考えられる。なお、筆者らは MAT4 に対応した IMS の性能評価を行っており、文献 9) でその詳細を述べている。

5.2 モバイルノード

MAT4 を Linux ホストに実装し、通信性能を評価した。この実験で使用したノードとネットワーク機器の諸元を表 3 と表 4 に示す。実験において IMS は IMS_{MN} と IMS_{CN} を 1

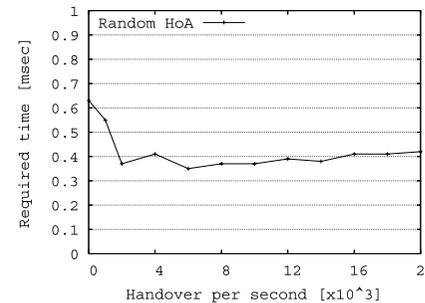


図 5 MAT6 のハンドオーバー処理時間
Fig. 5 Handover processing time of MAT6.

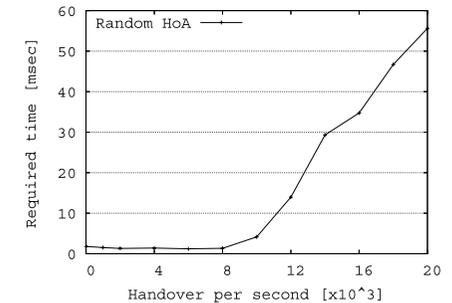


図 6 MAT4 のハンドオーバー処理時間
Fig. 6 Handover processing time of MAT4.

表 3 モバイルノード測定時の各ノードの諸元

Table 3 Node specifications in mobile node measurement.

Node	Hardware	CPU	RAM	OS
MN, CN	HP Mini 5101	Intel Atom N280 1.66 GHz	2 GB	Debian GNU/Linux 5.03 i386 linux-2.6.27.24 with MAT
IMS・DNS	HP Compaq 6005 Pro	AMD Athlon II X2 B24 3 GHz	4 GB	CentOS 5.5 i386 on VMware Server 2.0.2
Sniffer PC	Toshiba dynabook SS	Intel Pentium-M 1.1 GHz	512 MB	CentOS 5.5 i386 linux-2.6.30.10

表 4 モバイルノード測定時の各ネットワーク機器の諸元

Table 4 Network switch specifications in mobile node measurement.

Switch	Vendor	Model	Link Speed
L3SW・DHCP	Alaxala	AX3630S-24T	1 Gbps
L2SW	Allied Telesis	CentreCOM GS916S	1 Gbps
Repeater HUB(1)	Allied Telesis	CentreCOM MR415T	10 Mbps
Repeater HUB(2)	Net Agent	IDS-HUB	100 Mbps

台で兼用している。

(1) スループット測定

スループット測定に使用した機器の構成を図 7 に示す。ping コマンドにより、MAT4 あり・なしの場合において MN-CN 間の RTT (Round Trip Time) を計測し、MAT4 の処理による遅延時間を測定した。ペイロードサイズ 36 バイトの ICMP パケット (全長 64 バ

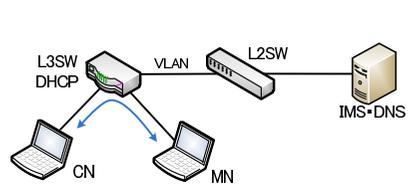


図 7 スループット測定時のネットワーク構成
Fig. 7 Throughput measurement system.

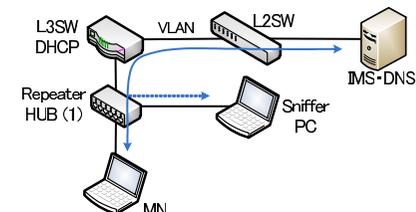


図 8 シグナリング動作時間の測定構成
Fig. 8 Signaling measurement system.

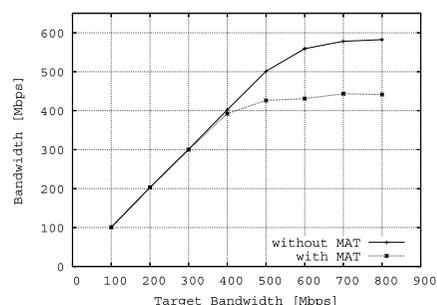


図 9 MN-CN 間のスループット
Fig. 9 Throughput between MN and CN.

表 5 MN-CN 間の RTT
Table 5 RTT between MN and CN.

RTT [msec]	min	avg	max
MAT4 なし	0.265	0.412	0.477
MAT4 あり	0.382	0.464	0.644

イトの IP パケット) を 100,000 個フラディングさせることにより, RTT の最小・平均・最大値を求めた。表 5 に測定値を示す。MAT4 ありの場合, 平均値を見ると 0.052 ミリ秒のわずかな増加となっている。

また, Iperf^{*1}により, MAT4 あり・なしの場合において 1,470 バイトの UDP データグラムを用いてスループット測定を行った。1 回の測定時間を 10 秒とし, 10 回の計測による平均値を求めた。測定結果を図 9 に示す。

MAT4 ありの場合, 400 Mbps までは MAT4 なしの環境と同程度のスループットを実現している。予備実験では, Iperf が CPU を 100%使いいきるアプリケーションであること, vmstat コマンドで MAT4 ありの場合は通信時にシステムの CPU 占有時間が大きくなることを確認しており, 通信負荷と CPU 負荷を同時にかかる用途では図 9 のようにスループット

*1 <http://sourceforge.net/projects/iperf/> (version 2.0.4)

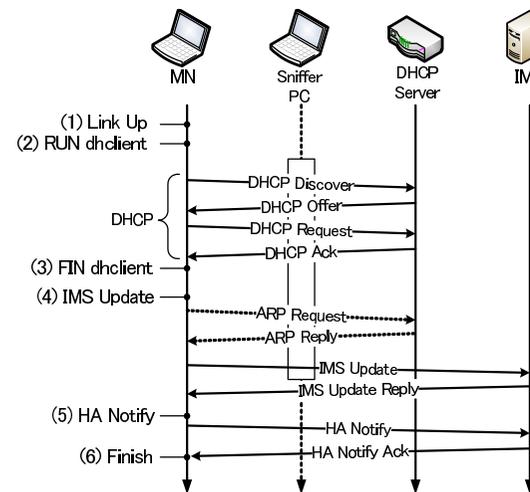


図 10 リンクアップ後のシグナリング
Fig. 10 Signals on link-up event.

トが低下する傾向がある。

(2) リンクアップから通信可能になるまでの所要時間測定

リンクアップからインタフェースが利用可能になるまでの時間を図 8 に示す構成で測定した。測定値は, シングルインタフェースでハンドオーバーする際の通信途絶時間の目安, および, 複数インタフェース使用時に 1 つのインタフェースが利用可能になる時間の目安となる。実験では, MN 上で MAT4 の制御デーモンプログラム (以降, mat4d) にイベントごとのタイムスタンプを表示させると同時に, スニファPC でネットワーク上を流れる DHCP パケットの経過時間を測定した。具体的なシグナリング動作を図 10 に示す。パケットキャプチャには Wireshark^{*2}を使用した。各ステップでの所要時間について, 表 6 に 10 回の試行における最小, 平均, 最大値を示す。

mat4d は OS 標準の dhclient を使用しており, 表 6 の DHCP は dhclient コマンドの実行時間 [(2) to (3)] に含まれている。dhclient コマンドは, 内部でランダム秒数の待ち時間の後に DHCP Discover を送信しているため約 1 秒から 3.5 秒を要しているが, 実質的な

*2 <http://www.wireshark.org/> (version 1.0.11)

表 6 シグナリング動作の概算時間
Table 6 Estimate time of each signal.

Steps	Required time [msec]		
	min	avg	max
(2) to (3) (DHCP)	999.5 (4.0)	2058.7 (11.7)	3494.2 (35.1)
(3) to (4)	0.3	0.4	0.5
(4) to (6)	4.0	13.2	24.7

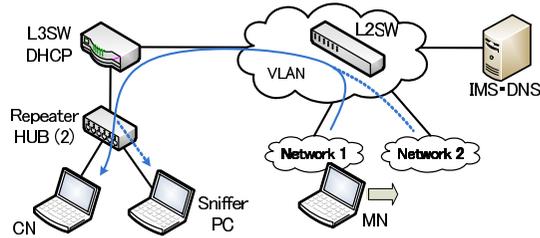


図 11 ハンドオーバー実験構成

Fig. 11 Structure of handover experiment.

DHCP にかかる時間は平均 11.7 ミリ秒である。

(3) ハンドオーバー時のパケット到達遅延時間測定

MAT4 のプロトコル仕様では、MN が CN へパケットを送信する際に HA Notify Ack を待つか否かを実装時に選択することができる。実験では、HA Notify などの応答時間をより明確にするために、HA Notify Ack を待つ仕様で実装して評価を行った。MN に USB ギガビットイーサネットアダプタ (PLANEX GU-1000T) を追加し、有線 LAN によるデュアルインタフェースによるハンドオーバーを行った。図 11 に構成図を示す。測定では Iperf を用いて、1,470 バイトの UDP パケットで 10Mbps のトラフィックを発生させた。実験では有線 LAN を使用しているが、遅延とジッタのあるネットワークでの使用も想定して、MN 上で tc コマンドを用いて仮想的にネットワーク遅延を生じさせた場合の測定も行った。tc コマンドは iproute2 に含まれ、2.6 系 Linux の netem カーネルモジュールと連携して、遅延とパケットロスなどを発生させることができる。tc コマンドにより 100 ミリ秒の遅延と 20 ミリ秒のジッタを発生させた。

以下では説明のために、インタフェース名を *eth0*、遅延・ジッタ付きを *eth0delay* のように表記する。MN から CN へ ping コマンドで 10,000 個の ICMP パケットを送信し、RTT

表 7 各インタフェースの RTT 特性
Table 7 RTT characteristics of interfaces.

RTT [msec]	min	avg	max	mdev
<i>eth0</i>	0.395	0.440	0.604	0.021
<i>eth1</i>	0.297	0.392	0.491	0.053
<i>eth0delay</i>	80.480	100.673	120.488	11.598
<i>eth1delay</i>	80.385	100.183	120.491	11.596

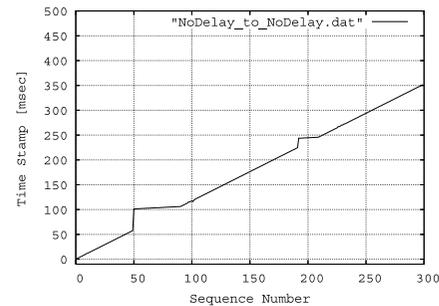


図 12 *eth0* から *eth1* へのハンドオーバー
Fig. 12 Handover from *eth0* to *eth1*.

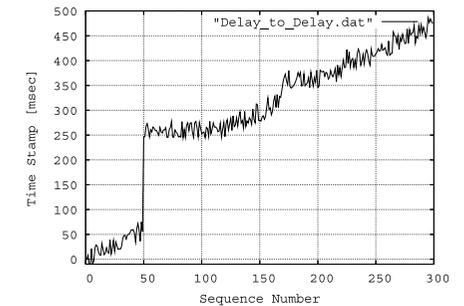


図 13 *eth0delay* から *eth1delay* へのハンドオーバー
Fig. 13 Handover from *eth0delay* to *eth1delay*.

の最小、平均、最大、平均偏差を測定した。結果を表 7 に示す。

パケット到達遅延の測定は複数回実施し、どの試行においても同じような傾向を示していること、およびハンドオーバーによるパケットロスが生じていないことを確認している。測定結果の例を図 12、図 13、図 14、図 15 に示す。図中の X 軸はシーケンス番号であり、図間比較のために 50 番と 51 番でハンドオーバーしているように調整してプロットしている。また、Y 軸は最初のシーケンス番号のパケットを受け取ってからの経過ミリ秒を表している。図 13 から図 15 の単調増加ではない区間は、シーケンス番号どおりにパケットが到着していないことを意味している。ハンドオーバー時の HA Notify は、図 10 の IMS Update 後に新しいインタフェースから出力される。ネットワーク遅延がほとんどなく、最も条件の良い図 12 でも約 40 ミリ秒の到達遅延が生じる結果となった。キューに溜ったパケットの送信に約 40 ミリ秒かかっていると思われる。遅延のあるインタフェース間でのハンドオーバーを想定した図 13 では、約 180 ミリ秒の遅延が生じた。図 14 は、遅延のないネットワークから遅延のあるネットワークへのハンドオーバーを想定しており、実測で 300 ミリ秒程度

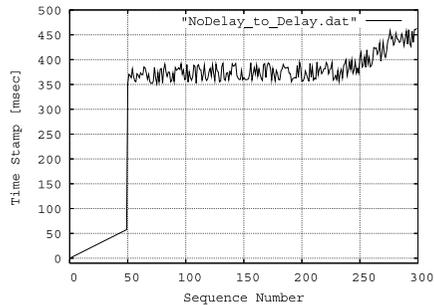


図 14 $eth0$ から $eth1_{delay}$ へのハンドオーバー
Fig. 14 Handover from $eth0$ to $eth1_{delay}$.

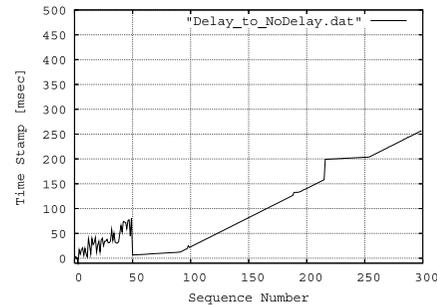


図 15 $eth0_{delay}$ から $eth1$ へのハンドオーバー
Fig. 15 Handover from $eth0_{delay}$ to $eth1$.

の遅延が測定された。図 15 はその逆の状況を想定している。図 13 と図 14 では、キューにあるパケットが多いためキューから排出する時間が長くなっている可能性があること、tc コマンドにより MN 上で仮想的に遅延を発生させているためキュー排出に時間がかかっている可能性がある。

5.3 考 察

MAT4 ではプロトコルに HA Notify と HA Query を追加しており、5.1 節で IMS 側でのハンドオーバーの処理負荷が高くなることが確認された。これは MAT4 でシグナリングに IP オプションを使用しないこととのトレードオフといえる。

5.2 節 (1) の測定結果より、MN はトンネルオーバーヘッドのないスルーブットを実現できている。MAT4 は最適経路で通信できることも加えると、VoIP アプリケーションに強いといえる。また (2) のリンクアップしてからインターフェースが利用可能になるまでの時間では、MAT6 からのオーバーヘッドは IMS との 1 往復分程度であり、むしろ dhclient の実行時間に大きく依存するといえる。そして (3) のハンドオーバー時のパケット到達遅延時間は、MAT4 で追加した HA Notify が影響している。MN 側で HA Notify Ack を待つ実装で実験したところ、パケット到達遅延は HA Notify の所要時間以上の遅延時間が生じる傾向にあり、MN の IMS との RTT に大きく左右されるといえる。一方で、MN が HA Notify Ack を待たずに CN にパケットを送った場合、条件によっては MAT6 のように遅延が発生しない¹⁰⁾ と予想される。CN は通信中、定期的に IMS_{MN} から MN のマッピング情報を取得しており、十分な時間間隔を開けてハンドオーバーすれば、CN- IMS_{CN} 間の HA Query は発生しない。逆に、ハンドオーバーの時間間隔が短いときは、CN が HA Query に成功す

るまで CN 側でキューイングによる遅延が発生する可能性がある。HA Notify 時における MN 側での送信キューの有無は、その影響について今後調査する余地があるといえる。

6. おわりに

本稿では、MAT を IPv4 ネットワークで使用できるように具体的にプロトコルの設計を行い、MAT4 のプロトタイプを Linux に実装、その性能評価を行った。実際に MAT4 を設計・実装すると、シグナリングや手順の増加などにより、MAT6 にはないオーバーヘッドが発生することが分かった。MN 側のハンドオーバー時のオーバーヘッドは理論的に HA Notify の実行時間であり、また実測において HA Notify の影響で IMS 側での処理負荷が増大することを確認した。

MAT4 は MN と CN がいずれも MAT4 に対応しなければならないという制約があるものの、Mobile IP の問題点である冗長な通信経路とアーキテクチャ上の単一障害点、モバイルノードの IP アドレスがネットワーク上の位置を正しく表していないという問題を解決している。さらに、MAT4 は Mobile PPC が対応していない UDP のようなコネクションレス型の通信においても移動透過通信を行うことができる。

MAT4 を本格的に運用していくためには、コンピュータやネットワークに不慣れな人でも導入しやすいようにしなければならないと考えている。今後は、こうした開発とは別の非技術的な部分にも力を入れるとともに、MAT4 対応ノードを増やすためにも、Linux や Windows だけではなく、たとえば Android などにも対応するなど、MAT4 のマルチプラットフォーム化を進めていきたいと考えている。

謝辞 本研究に関して、日頃より MAT に関する議論にご参加いただいている広島大学情報メディア教育研究センター関係者、広島市立大学インターネット工学研究室、株式会社ディアイティ、ネットワンシステムズ株式会社の関係各位に心より感謝いたします。なお、本研究の一部は、日本学術振興会科学研究費補助金 (20300029) および、総務省戦略的情報通信研究開発推進制度 (SCOPE-地域 ICT, 082308001) の支援を受けて実施しています。ここに記して謝意を表します。

参 考 文 献

- 1) Perkins, C. (Ed.): IP Mobility Support for IPv4, RFC 3344, IETF (2002).
- 2) 竹内元規, 鈴木秀和, 渡邊 晃: エンドエンドで移動透過性を実現する Mobile PPC の提案と実装, 情報処理学会論文誌, Vol.47, No.12, pp.3244-3257 (2006).

- 3) Johnson, D., Perkins, C. and Arkko, J.: Mobility Support in IPv6, RFC 3775, IETF (2004).
- 4) Ishiyama, M., Kunishi, M., Uehara, K., Esaki, H. and Teraoka, F.: LINA: A New Approach to Mobility Support in Wide Area Networks, *IEICE Trans. Communication*, Vol.E84-B, No.8, pp.2076–2086 (2001).
- 5) 相原玲二, 藤田貴大, 前田香織, 野村嘉洋: アドレス変換方式による移動透過性インターネットアーキテクチャ, *情報処理学会論文誌*, Vol.43, No.12, pp.3889–3897 (2002).
- 6) 寺岡文男: インターネットにおけるモバイル通信プロトコルの標準化動向, *電子情報通信学会論文誌*, Vol.J84-B, No.10, pp.1746–1754 (2001).
- 7) 岸場清悟, 藤田貴大, 田島浩一, 西村浩二, 前田香織, 相原玲二: MAT におけるマッピング情報管理サーバ冗長化に関する性能評価, *電子情報通信学会技術研究報告 IA, インターネットアーキテクチャ*, Vol.106, No.309, pp.19–24 (2006).
- 8) Droms, R.: Dynamic Host Configuration Protocol, RFC 2131, IETF (1997).
- 9) 岩田裕貴, 森廣勇人, 関 顕生, 前田香織, 井上博之, 相原玲二, 岸場清悟: 移動透過アーキテクチャMAT の IPv4 対応とその性能評価, *電子情報通信学会技術研究報告 IA, インターネットアーキテクチャ*, Vol.109, No.299, pp.7–12 (2009).
- 10) 相原玲二, 藤田貴大, 岸場清悟, 田島浩一, 西村浩二, 前田香織: 常に最適経路で通信を行う移動透過アーキテクチャMAT の性能評価, *インターネットカンファレンス 2006 論文集*, Vol.2006, pp.13–20, 日本ソフトウェア科学会 (2006).

(平成 22 年 6 月 14 日受付)

(平成 22 年 12 月 1 日採録)



関 顕生 (学生会員)

2009 年広島大学工学部第二類 (情報系) 卒業。現在, 同大学大学院総合科学研究科博士課程前期在学中。コンピュータネットワーク, IP モビリティに関する研究に従事。



岩田 裕貴

2009 年広島市立大学情報科学部情報メディア工学科卒業。現在, 同大学大学院情報科学研究科博士前期課程在学中。IP モビリティに関する研究に従事。



森廣 勇人

2008 年広島市立大学情報科学部情報メディア工学科卒業。現在, 同大学大学院情報科学研究科博士後期課程在学中。IP モビリティに関する研究に従事。



前田 香織 (正会員)

1982 年広島大学総合科学部卒業。同大学工学部助手, (財)放射線影響研究所技術員, 広島市立大学情報処理センター助教授を経て, 現在, 広島市立大学大学院情報科学研究科教授。博士 (情報工学)。コンピュータネットワーク, マルチメディア情報通信に関する研究に従事。電子情報通信学会, 教育情報システム学会各会員。



近堂 徹 (正会員)

2001 年広島大学工学部第二類 (電気系) 卒業。2006 年同大学大学院工学研究科博士課程修了。現在, 広島大学情報メディア教育研究センター助教。博士 (工学)。ネットワークコンピューティング, リアルタイムマルチメディア通信, QoS 保証技術に関する研究に従事。電子情報通信学会会員。



岸場 清悟 (正会員)

1988年京都大学理学部卒業。1994年同大学大学院理学研究科博士後期課程単位取得退学。現在、広島大学情報メディア教育研究センター助教。京都大学博士(理学)。主として分散システム運用およびモバイルネットワークアーキテクチャに関する研究に従事。日本物理学会、日本流体力学学会各会員。



西村 浩二 (正会員)

1989年広島大学工学部第二類(電気系)卒業。1991年広島大学大学院工学研究科博士課程前期修了。広島大学総合情報処理センター助手、同大情報メディア教育研究センター准教授等を経て、2011年より同教授。博士(工学)。コンピュータネットワークの運用管理、移動透過通信、情報セキュリティに関する研究に従事。電子情報通信学会会員。



相原 玲二 (正会員)

1981年広島大学工学部第二類(電気系)卒業。1986年同大学大学院博士課程修了。同大学助手、同大学集積化システム研究センター助教授を経て、現在、同大学情報メディア教育研究センター教授。工学博士。コンピュータネットワークに関する研究に従事。電子情報通信学会、IEEE Computer Society、IEEE Communications Society 各会員。