

# 移動透過通信アーキテクチャ用アドレス管理サービスの動的 DNSによる実装

大島 史也<sup>1</sup> 前田 香織<sup>1</sup> 相原 玲二<sup>2</sup> 近堂 徹<sup>2</sup>

**概要:** 移動透過通信アーキテクチャ MAT は、端末がネットワークを切り替えることによって端末の IP アドレスが変わっても、その変化をネットワーク層において吸収し、通信を継続することができる。このアーキテクチャでは、上位層で使用する端末固有の IP アドレス（ホームアドレス）と下位層で使用する移動ごとに変化する IP アドレス（モバイルアドレス）を、拡張したネットワーク層で変換する。これまで、変換に必要なアドレス情報を管理する専用サーバが必要であった。本稿では、このアドレス情報管理に一般的に使われている動的 DNS（Dynamic DNS）を利用することを提案する。DNS によるアドレス情報管理の実装方法について述べ、専用サーバで実装した場合との性能比較を通して、DNS によるアドレス情報管理の実現可能性を示す。

## Implementation of Address Management Service for an IP Mobility Architecture using Dynamic DNS

FUMIYA OBATAKE<sup>1</sup> KAORI MAEDA<sup>1</sup> REIJI AIBARA<sup>2</sup> TOHRU KONDO<sup>2</sup>

**Abstract:** One of IP mobility architectures, MAT (Mobility Support Architecture and Technologies) enables continuous connection even if an IP address of a terminal changes in handoff between networks by address mapping in a network layer. In MAT, we use two addresses for a mobile node; a permanent address (a home address) specifies the node's identity and a temporary address (a mobile address) represents the current location. The mapping function of these addresses is done in the network layer. Previously, MAT needs specialized servers to manage the address information for the address mapping. In this paper, we propose to use Dynamic DNS for this address management. We describe how to implement it by the DNS and show its feasibility through comparative performance evaluations with the case of the specialized server.

### 1. はじめに

移動端末の普及とともに移動透過通信への需要が高まっている。それに伴い、MIP[1] を始めとした様々な移動透過通信アーキテクチャが提案されている。著者らの開発する MAT (Mobility support Architecture and Technologies) [2] は、移動によって変わるアドレスであるモバイルアドレス (MoA) と端末を識別するためのアドレスであるホームアドレス (HoA) を、ネットワーク層でアドレス変換す

ることによって端末の移動透過通信をサポートする。このとき、アドレス変換に用いる IP アドレスの対応付けは専用のサーバ (IMS:IP address Mapping Server) で管理している。これにより、アドレス変換の一意性の確保やネットワーク移動時のハンドオーバーの迅速化を実現している。しかしながら、MAT の機能拡張に柔軟に追従できる反面、アドレス情報同期や認証機構などについても独自プロトコルによる定義が必要となり、専用サーバとしての設定や運用上の課題もある。これらが普及促進を難しくする一因となっている。

そこで本研究では IMS が果たしているアドレス管理機能 (以降、IMS 機能) の実装に DNS を用いることを提案する。動的 DNS (以降、DDNS[3]) を用いてアドレス情報

<sup>1</sup> 広島市立大学大学院情報科学研究科  
Graduate School of Information Sciences, Hiroshima City University

<sup>2</sup> 広島大学情報メディア教育研究センター  
Information Media Center, Hiroshima University

を管理することにより、IMS で課題となっていた同期や認証機構などに DNS プロトコルの枠組みをそのまま利用することができる。その一方で、DNS が持っているキャッシュ機構によってハンドオーバー時の情報更新の即時性が低下するなどの懸念もある。本稿ではアドレス管理サービスを DDNS によって実装することを提案し、性能評価について述べる。

本稿の構成は以下のとおりである。まず 2 章で移動透過通信におけるアドレス管理の観点から、移動透過通信アーキテクチャの違いについて述べる。次に、3 章で提案する DNS によるアドレス管理サービスの実装について述べ、DDNS のアドレスによる移動端末の IP アドレスの管理方法と、キャッシュサーバを介さないレゾルバの処理手法について述べる。4 章で提案方式と従来方式の比較評価について示し、5 章でまとめと今後の課題について述べる。

## 2. 移動透過通信におけるアドレス管理

### 2.1 移動透過通信で使用するアドレス

移動透過通信では、IP アドレスの役割を端末を一意識別するための端末識別子と、位置情報を表すロケータの 2 つに分離し、それぞれに別のアドレスを割り当てる。通常の TCP/IP では、端末識別子とロケータに同じ IP アドレスを用いているため、端末がネットワークを移動すると、端末識別子、ロケータともに同時に変更となり通信は途絶する。移動透過通信を実現するためには、2 つのアドレスに分離した上で相互に変換することで、移動時の通信継続性を確保する必要がある。また、端末のネットワーク上の位置を常に監視し、ロケータと端末識別子の対応関係を管理する機能が必要である。現在、移動透過通信を実現するために様々なアーキテクチャが提案されている。

### 2.2 MIPv6

Mobile IPv6 (MIPv6) [4] は、Mobile IP (MIP) [1] を IPv6 拡張したアーキテクチャであり、両者とも IETF で標準化されている。移動端末 MN (Mobile Node) の端末識別子であるホームアドレス HoA (Home Address) と、ロケータである気付けアドレス CoA (Care-of-Address) の二つのアドレスを持つ。HoA の属するネットワークであるホームリンク上に存在するホームエージェント HA (Home Agent) では、バインディングと呼ぶこの二つのアドレスの対応を常に把握し、HoA 宛てのパケットを MN の代わりに受信し、CoA 宛に IP ヘッダでカプセル化し、MN に向けて転送する。MN は HA からパケットを受け取り、通信相手端末 CN (Correspondent Node) に直接パケットを送信する。このように、MN の位置に関係なく HoA 宛にパケットを送信すれば HA が CoA 宛に転送するため、MN が移動した場合も継続して通信が可能になる。

しかし、MIP では HoA 宛のパケットを CoA 宛に転送

するために IP ヘッダでカプセル化する IP-in-IP トンネルを用いており、トンネリングによるオーバーヘッドが存在する。また、MIP では必ず HA を経由する冗長な経路を辿る三角経路問題が存在する。MIPv6 では経路最適化機能が追加され、冗長な経路を通らずに通信を行うことが可能になったが、最適経路での通信を行うためには MN と CN の間に SA (Security Association) を確立する必要がある。SA を確立するためには公開鍵インフラストラクチャが必要となるが、公開鍵インフラストラクチャは整備されていない。このため、依然として HA を経由する必要がある。実質的には MIPv6 も冗長経路による通信を行う [5]。また、HA を必ず経由するため HA が一点障害になることも問題点として挙げられる。

### 2.3 LIn6

LIn6[6] はアドレス縮退モデルによって端末識別子とロケータを分離し、移動透過性を保証する。LIn6 は 128bit からなる IPv6 アドレスのうち、下位 64bit をホスト固有の端末識別子として固定する。ネットワーク層より上のレイヤでは、アドレスの上位 64bit を IPv6 互換プレフィックスと呼ぶ固有値に差し変えた IPv6 互換ノード識別子を用い、ネットワーク層より下のレイヤでは上位 64bit を MN が今いるネットワークのネットワークプレフィックスに差し変えた縮退アドレスを用いる。これにより移動によって MN の IP アドレスが変わった場合も、CN のネットワーク層より上のレイヤでは MN 固有のアドレスである IPv6 互換ノード識別子のまま通信を行い、CN のネットワーク層より下のレイヤでは MN の移動先のネットワークプレフィックスに差し変えた縮退アドレスによって最適経路で通信を行うことができる。

MN の端末識別子とロケータの関係は binding と呼ばれ、BA (Binding Agent) によって管理されている。MN はネットワークを移動した際に自身の binding を管理する BA に移動先のネットワークの位置指示子を通知する。MN/CN の binding を管理する BA の情報は DNS によって管理されている。MN が CN と通信を行う際、まず DNS から CN の binding を管理する BA を取得し、BA から最新の binding を取得することで CN が今いるネットワークの位置指示子を元にした縮退アドレスを生成する。

LIn6 は MIPv6 の持つ冗長経路問題や一点障害を解決しているが、アドレス空間の制約からネットワークモビリティへの拡張は困難であり、実際に LIn6 のネットワークモビリティ拡張である LIn6-NEMO[7] における通信はマッピングエージェントと呼ばれる中継ノードを経由し経路最適化されない。

### 2.4 MobilePPC

MobilePPC (Mobile Peer to Peer Communication) [8]

は両エンド端末においてアドレス変換処理を行う移動透過通信技術である。MN はホスト名と IP アドレスの関係を DDNS[3] によって最新に保っており、MN と通信を開始する際はホスト名から最新の IP アドレスを取得し通信を確立する。MN と CN が既に通信を確立しているとき、MN が別のネットワークに移動すると、MN は CN に向けて新しい IP アドレスと移動前に確立していた接続に関する通信識別子を含む CU (CIT Update) パケットを送信する。CU パケットを受信した CN は自身の CIT (Connection ID Table) を更新し、IP 層で移動前の IP アドレスと移動後の IP アドレスを相互にアドレス変換する。これにより、上位層からは移動前の IP アドレスと通信を継続しているように見え、移動後も透過的に通信を行うことができる。

MobilePPC では、ロケータとして用いていた IP アドレスが MN の移動によって識別子としての IP アドレスに切り替わると考えることができる。また、これらのアドレスの管理や相互変換は端末上で行われている。MobilePPC ではエンドツーエンドでアドレス変換処理を行うため最適経路で通信でき、トンネリングによるオーバーヘッドも存在しないため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 が直接送り合うため、専用の装置が必要なく MIP の HA のような一点障害が存在しない。しかし、MobilePPC は MN と CN が直接 CU パケットを送信しあうため、ノード間での認証が必須となる。このため、通信開始時に鍵交換などの処理で費されるオーバーヘッドが大きい。更に、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。

## 2.5 MAT

MAT はロケータであるモバイルアドレス (MoA) と、端末識別子であるホームアドレス (HoA) をネットワーク層で変換することで移動透過通信を実現する。端末ごとに移動透過通信をサポートするホストモビリティを提供する MAT-HOST では、アドレスの変換は端末上で行われる。図 1 に MAT の通信におけるアドレス変換の概要を示す。端末間の経路は通常のルーティングと同じ最適な経路を辿るため、MIP や MIPv6 のような三角経路問題は発生しない。

MN が CN と通信を開始する際、または移動してアドレスが変更する際の MN、CN と各サーバのやりとりを図 2 に示す。MN は移動によってアドレス情報が変更された時、アドレス管理サーバである IMS (IP Address Mapping Server) に新しい MoA の情報を通知し、IMS のアドレス情報を更新する。図 2 のように MN の IMS を IMS (MN) と記す。なりすましを防止するため、IMS のアドレス情報の更新には共通鍵による認証が用いられる。更新が完了した後、MN は CN に移動が完了したことを通知するため

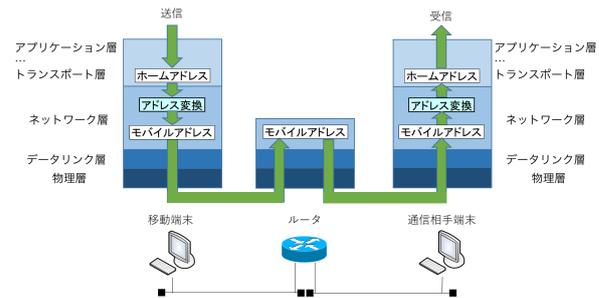


図 1 MAT のアドレス変換

Fig. 1 Address translation of MAT

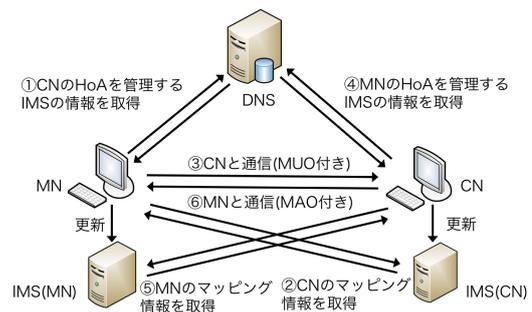


図 2 各サーバ, MN, CN 間の通信フロー

Fig. 2 Overview of communication flow among servers, MNs and CNs

に、拡張オプションヘッダである MUO (Mapping Update Option) を付与したパケットを送信する。更新通知 (MUO) を受信した CN は MN のアドレス情報を IMS (CN) から取得し、自身のアドレス変換テーブル (IMT) を更新する。その後、CN は更新完了の ACK として拡張オプションヘッダである MAO (Mapping Ack Option) を付与したパケットを MN に送信する。

IMS が動作するサーバは IMS レコードを扱えるよう拡張した DNS によってその位置が管理されている。HoA の逆引きのゾーンに IMS レコードを設置することで、MN は DNS から HoA のアドレス情報の管理を行っている IMS が動作するサーバを知ることができる。また、IMS レコードは一つのゾーンに複数設置することができるため、一つの HoA に対して複数台の IMS が動作するサーバを設置することで可用性の向上が期待できる。

## 3. DNS によるアドレス情報管理

### 3.1 IMS と DNS の課題

本稿では、DNS および IMS の機能や動作を「DNS 機能」および「IMS 機能」と呼ぶ。一方、BIND[9] など DNS 機能を実装したソフトウェアを「DNS プログラム」または「DNS」、IMS 機能を実装したソフトウェアを「IMS プログラム」または「IMS」と表記する。

先述の通り、IMS は MAT の HoA と MoA のアドレス



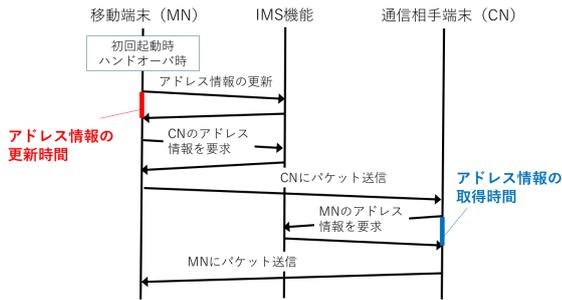


図 4 ハンドオーバー時のメッセージフロー  
Fig. 4 Message flow in handover

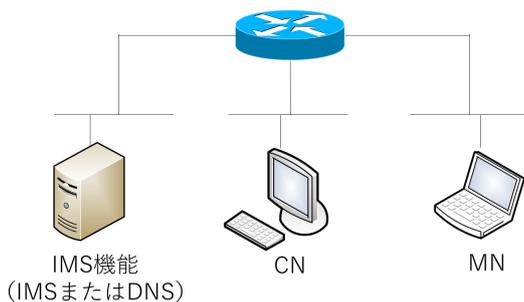


図 5 アドレス更新に関する時間の測定用の実験構成  
Fig. 5 Experiment configuration for time evaluation on IP address update

表 1 測定に用いた MN/CN の性能  
Table 1 Specification of MN/CN

	MN/CN
OS	Linux (CentOS6.7 i386 版)
CPU	Intel(R) Core(TM)2 Duo CPU U9600 1.60 GHz 2 コア
メモリ	4GB
カーネル	2.6.32.16
コンパイラ	gcc-4.4.7

ローは従来方式も提案方式も同じだが、提案方式の場合、DNSを用いることでアドレスの更新時間や取得時間が増加することが予想される。従って、これらの時間を計測した。

#### 4.1.1 実験環境と実験方法

測定に用いたネットワーク構成図を図 5 に示す。また、測定に用いた MN/CN と IMS 機能が動作するサーバ (IMS/DNS) の仕様をそれぞれ表 1 と表 2 に示す。IMS/DNS には 10 個のエントリが格納された状態で測定を行った。計測は端末の MAT デーモン内に gettimeofday 関数を挿入し、IMS 機能への通信開始から IMT への書き込みが完了するまでの経過時間を取得している。

#### 4.1.2 実験結果

アドレスの取得時間はアドレス情報の管理を従来通り IMS で行った場合と IMS 機能が動作する DNS で行った場

表 2 IMS 機能が動作するサーバの性能

Table 2 Server(IMS/DNS) specification

	IMS/DNS
OS	Linux (CentOS7.2 x86_64 版)
CPU	Intel(R) Core(TM) i5 CPU M 520 @ 2.40GHz 4 コア
メモリ	6GB
カーネル	3.10.0-327.13.1
コンパイラ	gcc-4.8.5
DNS サーバ	BIND-9.10.3-P3
BIND のスレッド数	4
IMS のスレッド数	5
ファイルシステム	XFS

表 3 アドレス情報の取得時間

Table 3 Time of acquiring address information

	取得時間 [ms]
従来方式 (IMS)	1.665
提案方式 (DNS)	5.163

表 4 アドレス情報の更新時間

Table 4 Time of updating address information

	更新時間 [ms]	
従来方式	IMS 1.751	
提案方式	DNS(IO バリア有効時)	172.790
	DNS(IO バリア無効時)	27.635

合とでそれぞれ 10 回計測を行い、その平均を表 3 に示す。同様にアドレスの更新時間も、従来方式と提案方式とでそれぞれ 10 回計測を行い、その平均を表 4 に示す。表 4 には IO バリアを無効にした場合の結果も示す。これは更新時間を測定するうえで DNS 機能の動的更新にかかる時間にファイルシステムの機能の 1 つである IO バリアが大きく影響を及ぼすからである。

## 4.2 DNS (権威サーバ) の処理能力の測定

通常の DNS 機能では、名前解決において端末はキャッシュサーバに対して問い合わせ、権威サーバに直接問い合わせることはない。しかし、提案方式では最新のアドレス情報を得るために IMS 機能をもつ DNS (権威サーバ) へ端末が直接問い合わせ、DNS 機能の権威サーバに比べ IMS 機能をもつ権威サーバの方が負荷が集中することが想定される。そこで、提案方式ではどれくらいアドレス情報の問い合わせを処理できるかを調べ、従来方式の IMS 専用サーバと比較した。

### 4.2.1 実験環境と実験方法

測定環境は図 6 のとおりで、問い合わせ端末に IMS 機能つきの DNS に向けて、アドレス情報を問い合わせる (DDNS の TXT レコードの送出) プログラムを入れ、問い

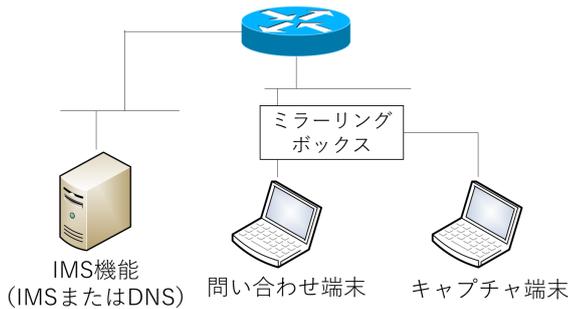


図 6 実験構成

Fig. 6 Experiment configuration for server performance evaluation

表 5 問い合わせ端末の仕様

Table 5 Specification of a measurement terminal

	問い合わせ端末
OS	Linux (Gentoo Linux amd64)
CPU	Intel(R) Core(TM) i7-4600U CPU @ 2.10GHz 4 コア
メモリ	16GB
カーネル	4.3.2-gentoo
コンパイラ	gcc-4.9.3

合わせに対するレスポンス数を測定した。IMS 専用サーバのときも相当する問い合わせメッセージを送り、レスポンス数を測定した。

問い合わせ端末の仕様を表 5 に示す。IMS 機能が動作するサーバは表 2 と同じものを使用した。また、iperf3 を用いた問い合わせ端末とサーバ間のスループットは 919Mbps であった。レスポンス数の測定は問い合わせ端末とサーバの間に設置したミラーリングボックスでミラーリングされたパケットをキャプチャ端末で測定した。問い合わせ端末はサーバに対し、16 個のプロセスから並列にアドレス情報の問い合わせ用のリクエストパケットを送信する。リクエストパケットは実際に IMS または IMS 機能が動作する DNS に問い合わせを行った時のパケットのペイロード部をコピーしたものを用いる。また、問い合わせるアドレス情報はサーバ上にエンタリが存在する 10 個の HoA から毎回ランダムに選択する。各プロセスは無限ループになっており、ループ内では問い合わせる HoA の選択、sendto/rcvfrom を行うことで、サーバにアドレス情報を問い合わせ続ける。IMS と IMS 機能が動作する DNS への問い合わせは同じ測定プログラムを用いており、ペイロード部と送信ポート番号のみ異なる。

#### 4.2.2 実験結果

IMS 機能が動作する IMS または DNS でそれぞれ 10 分間の測定を行い、送信されたリクエストパケット数とそのレスポンスパケット数、また 1 秒あたりのレスポンスパ

表 6 サーバのパケット処理能力の比較

Table 6 Performance comparison on processing packets

	リクエスト パケット [個]	レスポンス パケット [個]	1 秒あたりの処理 パケット [個/秒]
従来方式 (IMS)	52,254,174	52,254,162	86,449
提案方式 (DNS)	12,885,907	12,885,904	21,423

表 7 エントリ数の増加によるレスポンス時間

Table 7 Response time by increasing the number of entries

エンタリ数 [個]	10	100	1,000	10,000	100,000
応答時間 [μs]	IMS 699	IMS 689	IMS 687	IMS 689	IMS 684
	DNS 947	DNS 964	DNS 968	DNS 977	DNS 967

ケット数を処理パケット数として表 6 に示す。tcpdump のログから、IMS の測定では測定端末上で 0.4% のパケットが欠落していることが確認できている。

#### 4.3 管理エンタリ数とレスポンス時間への影響

MAT によってホストモビリティを実現する際は MN1 台毎のアドレス情報を IMS 機能により管理する。IMS 機能が動作するサーバ 1 台で複数の MN のアドレス情報を同時に管理することができる。1 台で管理するエンタリ数が増加した場合、MN や CN からのリクエストに対するサーバのレスポンス時間が増加することが懸念される。そこで、サーバ 1 台の管理エンタリ数を 10 個、100 個、1,000 個、10,000 個、100,000 個とした場合のレスポンス時間を測定し評価を行った。

##### 4.3.1 実験環境と実験方法

表 5 と同じ測定端末と表 2 と同じ IMS 機能が動作するサーバ (IMS/DNS) 1 台を同じサブネットに設置して測定を行った。計測はサーバにリクエストパケットを送り、そのレスポンスが返ってくるまでの時間を計測した。

##### 4.3.2 実験結果

IMS と IMS 機能が動作する DNS に 10 個、100 個、1,000 個、10,000 個、100,000 個のアドレス情報を登録し、管理数の範囲内でランダムに選んだ HoA を 100 回問い合わせた平均を表 7 に示す。

#### 4.4 考察

実験の結果から従来方式に比べて、提案方式の IMS 機能が動作する DNS の場合、表 3 からアドレス情報の取得時間が 3.5ms ほど、表 4 から IO バリア無効時は更新時間 (IO バリア無効時) が 25.9ms ほど増加するという結果が得られた。IO バリアが無効である場合であっても更新時間が 25.9ms 増加している原因は、MN で動作している MAT デーモンの実装が従来のものと異なるためだと考えられる。通信アプリケーションの用途ごとの通信品質の指

標を定めている ITU-T G.114[11] では、最も遅延が許されない音声通話アプリケーションにおいて許容できる遅延時間は 150 ms であるとしている。今回の測定結果はこの数値の範囲内であり、DNS によって IMS 機能を代替することが MN/CN 間の通信品質に及ぼす影響は、RTT などのその他の要因が大きくなると考えられる。

提案方式の場合の DNS の問い合わせ処理の能力に関しては、表 6 より、DNS では秒間 21,423 個のリクエストパケットを処理できることが分かった。通常、IMS 機能にアクセスがあるのは CN のアドレス情報が IMT に登録されていない初回通信時や、MN が移動したことでアドレス情報に変更された時である。このため、アクセスの多いサーバなどのアドレス情報を管理する IMS 機能であっても、問い合わせは高々数百 pps であり、今回測定したような数千 pps を越えるようなリクエストは通常考えられない。従来方式の IMS と比較すると IMS 機能が動作する DNS では 1 秒間に処理できるパケット数が約 4 分の 1 になっているが、これはパケットサイズの差とサーバの処理時間の違いによるものと考えられる。測定結果より、DNS では 10,000pps を越えるリクエストであっても処理可能である。

アドレス情報の数が増加した場合のレスポンス時間に関しては、表 7 より従来方式と提案方式で 300  $\mu$ s 位の差はあるものの、エントリ数が 100,000 個まではほぼ同じレスポンス時間であることがわかった。以上から、提案方式の DNS でアドレス情報を管理した場合も IMS と同等の性能で利用可能であることを示せた。

## 5. まとめ

本稿では、動的 DNS を用いて移動透過通信アーキテクチャの 1 つである MAT で必要な IP アドレス情報管理のプロトタイプシステムを実装した。DNS でアドレス情報を管理した場合の性能を実験によって従来方式の IMS と比較し評価した。実験の結果から、DNS でアドレス情報を管理することでアドレス情報の取得時間と更新時間が増加することが分かった。DNS の権威サーバ単体であっても通常の利用範囲の問い合わせであれば正常に処理できることを確認した。また、レスポンス時間が管理エントリ数に依存しないことを示した。

評価より IMS に比べて処理時間が増えるものの、一般的なサーバでアドレス情報を管理できる可能性を示した。一般的な DNS サーバを用いることで、IMS のように専用の設定や管理が必要といった運用上の問題が解決できることを述べた。

今回の実装では DNS のみでアドレス管理を行っているが、本提案は従来方式の IMS によるアドレス管理とも共存が可能である。IMS によるアドレス管理と混在させる場合、HoA の逆引きのゾーンに本実装で用いた TXT レコードの他に、従来の IMS レコードを追加すればよい。MN の実装

を変更する必要はあるが、これは今後の課題とする。

今後の予定として、キャッシュサーバに問い合わせることによって CN のアドレス情報を管理する権威サーバを探索する処理を実装する。

謝辞 本研究に関して議論し、ご助言を頂きました MAT プロジェクトの関係各位に感謝致します。

本研究の一部は日本学術振興会科学研究費助成金 16H02808, 15K00130 の支援を受けて実施しました。

## 参考文献

- [1] Perkins, C.: IP Mobility Support for IPv4, RFC 3344 (Proposed Standard) (2002).
- [2] 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).
- [3] Vixie, P., Thomson, S., Rekhter, Y. and Bound, J.: Dynamic Updates in the Domain Name System (DNS UPDATE), RFC 2136 (Proposed Standard) (1997).
- [4] Johnson, D., Perkins, C. and Arkko, J.: Mobility Support in IPv6, RFC 3775 (Proposed Standard) (2004).
- [5] 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. B, 通信, Vol. 84, No. 10, pp. 1746–1754 (2001).
- [6] ISHIYAMA, M., KUNISHI, M., UEHARA, K., ESAKI, H. and TERAOKA, F.: LINA: A New Approach to Mobility Support in Wide Area Networks (Special Issue on Internet Technology), *IEICE transactions on communications*, Vol. 84, No. 8, pp. 2076–2086 (2001).
- [7] BANNO, A., OIWA, T. and TERAOKA, F.:  $\chi$  LIN6-NEMO: A Network Mobility Protocol Based on LIN6(<Special Section>Internet Technology VI), *IEICE transactions on communications*, Vol. 89, No. 4, pp. 1070–1079 (2006).
- [8] 竹内元規, 渡邊 晃: モバイル端末の移動透過性を実現する Mobile PPC の提案, 情報処理学会研究報告モバイルコンピューティングとユビキタス通信 (MBL), Vol. 2004, No. 95, pp. 17–23 (2004).
- [9] Internet Systems Consortium: BIND, <https://www.isc.org/downloads/bind/>.
- [10] 関 顕生, 岩田裕貴, 森廣勇人, 前田香織, 近堂 徹, 岸場清悟, 西村浩二, 相原玲二: 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, 情報処理学会論文誌, Vol. 52, No. 3, pp. 1323–1333 (2011).
- [11] ITU-T Recommendation G.114: Transmission systems and media, general characteristics of international telephone connections and international telephone circuits; One-way transmission time (1996).