Vol.2010-MBL-52 No.5
2010/1/28

IPv6 における Mobile PPC の実現と評価

寺 澤 圭 史[†1]　鈴 木 秀 和[†1,†2]　渡 邊　晃[†1]

ユビキタスネットワークでは，通信中に端末が移動しながらでもコネクションを切断せずに通信継続が可能な移動透過性が必要となる．我々はエンド端末だけで移動透過性を実現するプロトコルとして Mobile PPC（Mobile Peer-to-Peer Communication）を提案している．Mobile PPC は現在 IPv4 スタックで実装されており，有効性も確認済みである．近年では IPv6 が普及し始めており，IPv6 ネットワーク環境において移動透過性も重要であることは明らかである．そこで，本稿では Mobile PPC の原理をそのまま IPv6 に適応させ，Mobile PPC の特徴を生かしたまま Mobile PPCv6 の実装を行った．IPv4 では DHCP によるアドレス取得に時間を要するため，ハンドオーバ時の通信断絶時間が非常に長くなってしまっていた．しかし，IPv6 では端末がアドレスを自分で生成するため，アドレスの取得に時間がかからない．IPv6 で課題となる移動検知と重複アドレスチェックについては独自の機能追加を行い通信断絶時間の短縮を試みた．Mobile PPCv6 の性能評価を行った結果，所定の移動透過性を実現でき，かつ既存技術に比べ高速なハンドオーバが可能なことが分かった．

The Realization of Mobile PPC in IPv6 Networks and its Evaluation

Keiji Terazawa,[†1] Hidekazu Suzuki[†1,†2]
and Akira Watanabe[†1]

In ubiquitous networks, mobility, the ability of keeping of mobile nodes, is the importan function. We have been studying the new technology called Mobile Peer to Peer Communication（Mobile PPC）that can achieve Mobility only with end nodes. Mobile PPC is implemented in IPv4 stack and also confirmed the effectiveness. In recent years, IPv6 networks is abount to spread, and the Mobility IPv6 network is finally become the key issue. So, in this paper, we have realized MPPCv6 applying the principle of Mobile PPC. In IPv4, it took a long time for the hand-over proccess due to the slow DHCP negotiation, however, it can b shortend due to IP address auto configuration. We device the function of the duplicate address detection and also modified the MPPC sequence. From the result of the performance evaluation, it can be proved that fast handover is possible in Mobile PPCv6.

1. はじめに

モバイル端末や公衆無線環境の普及に伴い，移動しながら通信を行いたいという要求が高まっている．しかし，IP ネットワークでは，通信中に端末がネットワークを移動することにより様々な問題が生じる．TCP/IP の通信では IP アドレスとポート番号で通信を識別しているため，IP アドレスが変化すると通信が切断してしまう．また，通信を開始しようとしても移動により IP アドレスが変化してしまうと，宛先が特定できず通信を開始できないという課題がある．これらの課題を解決するための機能を移動透過性と呼び，様々な方式が検討されている[1)]．

そこで，我々はこれまでエンドエンドで移動透過性を実現する通信プロトコルとして Mobile PPC（Mobile Peer to Peer Communication）[2)] を提案してきた．Mobile PPC（MPPC）は，エンド端末だけで移動透過性を実現でき，特別な装置は不要である．

一方，IPv4 グローバルアドレスの枯渇により今後 IPv6 が普及する．IPv6 では，IPv4 で用いられている NAT（Network Address Translator）が存在しないため，アドレスにグローバルやプライベートのような区別はない．すべての端末は一意性が保たれたグローバルユニークなアドレスを使用して，常にエンドエンドの通信が可能となる．そのような IPv6 ネットワークでは移動透過性を実現する場合においても，エンドエンド通信であることが望ましい．

IPv6 では Mobile IPv6（MIPv6）[3)] と呼ばれる移動透過性技術が基本技術の一つとして存在する．MIPv6 とは，IPv4 で移動透過性を実現する Mobile IP（MIP）[4)] をベースとして IPv6 へ対応させた技術である．MIPv6 では HA（Home Agent）と呼ぶ装置が必要で，一部の経路には IPsec[5)] を使用しなければならない．大まかに分類すると HA を用いたトンネルプロトコルであるが，冗長経路が発生してしまう．更に，様々な独自機能も備わっているため，大変複雑なシステムとなっており普及には至っていない．

MPPC は現在，IPv4 での実装・評価を終え，その有効性が証明されているが，その原理は IPv6 にもそのまま適応可能である．本稿では Mobile PPC の原理と特徴を活かしたまま，

†1 名城大学大学院理工学研究科
Graduate School of Science and Technology, Meijo University
†2 日本学術振興会特別研究員 PD
Research Fellow of the Japan Society for the Promotion of Science

Vol.2010-MBL-52 No.5
2010/1/28

IPv6 ネットワークでも移動透過性を実現する Mobile PPCv6（以後，MPPCv6）の実装を行い，その評価を行った．MPPC では DHCP によるアドレス取得や重複アドレスチェックに非常に多くの時間を要していた．IPv6 ではルータから広告されているネットワークプレフィックスから端末自身がアドレスを自動で生成するため，アドレス生成に要する時間が短い．しかし，重複アドレスチェックの処理時間について無視できないため，独自の実装方式を用いた．それに伴い MPPC のネゴシエーションも一部改善が必要なため，MPPCv6 のシステムにも追加実装を行う．さらに，独自でネットワークの移動を監視する機構を実装することで，ハンドオーバの検出にかかる時間を大幅に短縮した．

MPPCv6 の性能評価を行うために IPv6 無線ネットワークを構築して実験を行った．無線環境で MPPCv6 を用いて L2,L3 ハンドオーバを行った場合の各処理時間を測定して評価を行う..

以降，2. で Mobile IPv6 とその課題，3. で Mobile PPC 概要を説明する．4. で Mobile PPCv6 の実装，5. で性能評価を行い，6. でまとめる．

2. Mobile IPv6

既存技術として，IPv6 で標準搭載されている Mobile IPv6 をとりあげる．MIPv6 では MIP と同様に，通信開始時のネットワーク（ホームネットワーク）で取得するホーム・アドレス（HoA;Home Address）と訪問先ネットワークで取得する気付アドレス（CoA;Care of Address）の二つのアドレスを用いる．HoA は移動しても変化することがなく，アプリケーションは自端末のアドレスを常に HoA と認識する．また，ホームネットワークには HA が設置されており，移動後は HA を使用することで移動透過性を実現する．**図 1** に MIPv6 の概要を示す．ホームネットワークに存在する移動端末（MN;Mobile Node）が通信相手（CN;Correspondent Node）が通信中に異なるネットワークに移動した場合について示す．ホームネットワークに存在する MN は HoA を用いて CN と通信を開始する．通信中に MN が移動先ネットワークへ移動すると，新たに CoA を取得する．CoA を取得した MN は HA に対してバインディング通知（BU;Binding Update）を行い，HoA と CoA の関係が示されているバインディングキャッシュ（BC;Binding Cache）を更新する．このバインディング通知メッセージは盗聴やハイジャックを防止するために IPsec による保護が必須となっている．BC を更新後，MN は送信元 HoA のパケットを送信元 CoA のヘッダでカプセル化して HA に送信する．それを受け取った HA はパケットをデカプセルして CN へ転送する．CN から MN への通信は上記と逆方向トンネルを用いてパケットを転送する．以上のように HA を介した双方向トンネルを用いることにより移動透過性を実現するが，課題もある．

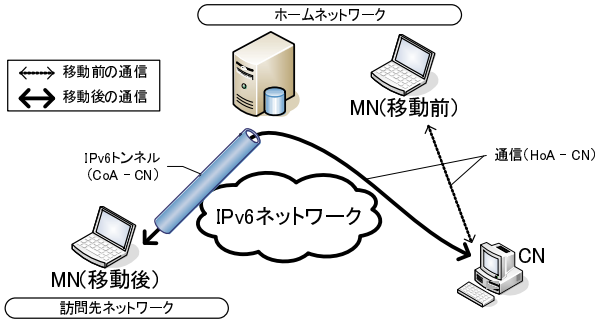

図 1 Mobile IPv6 の概要
Fig. 1 Overview of Mobile IPv6.

MIP のシステムは HA の設置が必須であるため，各ネットワークには HA の設置が前提となる．移動後の通信では HA を介した IPv6 双方向トンネルになるため，冗長経路と帯域消費が発生する．更に，CN も移動した場合，CN もホーム・ネットワークに設置された HA とトンネルを形成してしまう．そのため MN と CN 間の通信経路上で二つの HA を経由し，それぞれの HA との間でトンネルを形成しなければならない．

また，リアルタイムな通信には経路最適と呼ばれる機能が用いられる．経路最適化とは HA を介したトンネル経路ではなく，エンドエンドで通信を行う機能である．経路最適化を行うにはモビリティヘッダと呼ばれる IPv6 拡張ヘッダが必要となる．MN がパケットを送信する場合，送信元 HoA から送信元 CoA に書き換え，モビリティヘッダに HoA を記述して CN に送信する．CN が受信するときは IP 層で送信元 CoA からモビリティヘッダに記述されている HoA に変換してアプリケーションに渡す．パケットを受け取った CN は送信元を CoA から HoA に書き換えてアプリケーションへ渡す．CN からパケットを送信する場合は宛先のアドレスを CoA に書き換え，モビリティヘッダに HoA のアドレスを記述して送信する．パケットを受け取った MN は同様に宛先を HoA に書き換えてアプリケーションへ渡す．以上のような動作によりエンドエンドでの通信継続を可能とする．経路最適化を用いた場合，CN が移動した場合に発生した冗長経路はなくなるが，HA は必須であり，パケットのオーバヘッドも発生してしまう．

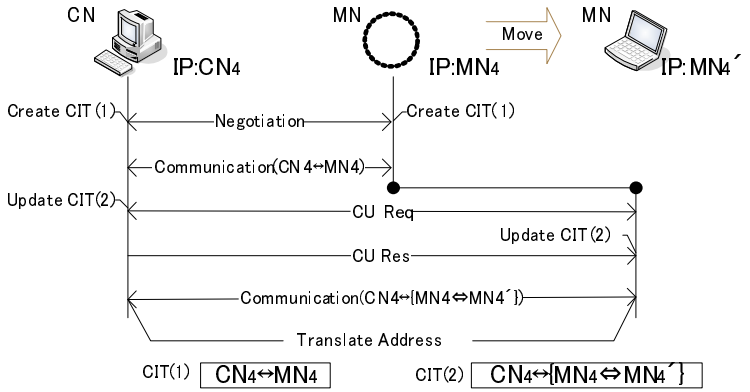

図 2 Mobile PPC の動作
Fig. 2 Sequence of Mobile PPC.

3. Mobile PPC の概要

本稿で用いる記号を以下のように定義する．

- A; 端末の IP アドレス
- $A \rightarrow B$; A から B への通信
- $A \leftrightarrow B$; A と B 間の通信
- $A \Leftrightarrow B$; A から B，または B から A へのアドレス変換

Mobile PPC はエンド端末だけで移動透過性を実現する通信プロトコルである．通信開始時における通信相手の IP アドレスの解決には DDNS（Dynamic Domain Name System）[6] を使用する．

アプリケーションに対してアドレスを隠ぺいするために MIPv6 ではカーネルでのトンネルやモビリティヘッダを用いていたが，MPPC ではカーネルに作成したアドレス変換テーブルを用いて行う．このような方式によりパケットヘッダのオーバヘッドが発生しない．

図 2 に MobilePPC のシーケンスを示す．通信開始に先立ち，Diffie-Hellman（以下 DH）鍵交換を用いて共通鍵を生成する．ネゴシエーション完了後，MN と CN には CIT（Connection ID Table）が IP 層に生成される．この時 CIT には通信開始時（移動前）のコネクション識別子 CID（Connection ID）が記録されている．通信中に MN がネットワークの移動するとアドレス自動生成により新しい IP アドレス MN_6' を取得し，CU（CIT Updata）ネゴシエーションを開始する．CU ネゴシエーションでは移動後の CID を直接通知し合うことでお互いの CIT に移動後の CID を追加する処理を行う．まず，MN は移動後の IP アドレス MN_6' を通知するために CU Request を CN に送信する．CU Request には共通鍵より生成した MAC（Mesage Authentication Code）が付加されており，盗聴やハイジャックを防ぐことができる．CN は CU Request の内容を認証後，自らの CIT を

$$CIT: \quad CN_6 \leftrightarrow \{MN_6 \Leftrightarrow MN_6'\} \tag{1}$$

のように更新する．次に， CN は MN に対して CU Response を送信する．MN は CU Response を認証後， (1) と同様に自らの CIT を更新する．以後は，更新された CIT の (1) の内容に従って，全ての通信パケットのアドレス変換を行うことにより，通信を継続することができる．

Mobile PPC は IPv4 スタックへの実装と評価を完了しており，その有用性が証明されている．MPPC の原理は IPv6 スタックにも同様にして適用可能である．MPPC の原理と特徴を生かしたまま IPv6 の実装を行った．

4. Mobile PPCv6 の実装

4.1 実装の概要

MPPC は現在メインシステム部分は IPv4 スタックに実装されており，アプリケーション層には鍵生成やインターフェースの監視などのデーモンが実装されている．カーネルに実装されているメインシステムの大部分は IPv6 スタックへそのまま移植が可能であるため，MPPC モジュールを IPv6 に対応させた．しかし，IPv4 では発生しなかった IPv6 特有の課題があるので，それらについては新たな実装を行った．

MPPCv6 で新たな実装を行ったのは NDP（Neighbor Discovery Protocol）による重複アドレスチェック（DAD;Duplicated Address Detection）に関わる部分である．NDP は IPv4 の ARP に相当する処理で，IPv6 では ICMPv6 の機能として定義されている．NDP の中でも DAD の処理時間が移動透過性の性能に大きく影響を与えることが分かった．DAD の処理時間を大幅に短縮する機構を MPPCv6 で独自に実装を行い，それに伴って移動通知処理である CU ネゴシエーションにも改良を加えた．また，端末の移動をアプリケーションで検知する実装を行った．これはルータから広告される Router Advertisement（RA）をトリガにして移動検知を行うと検知が遅れ，パケットロスが大量に発生するためである．

4.2 DAD 処理の短縮

4.2.1 DAD 処理概要

移動透過性を実現する技術では，ハンドオーバ時に迅速に通信を再開できることが望ましい．DAD は NDP の機能一つとして実装されており，ICMPv6 パケットをベースに動作する．IPv6 アドレスには 3 つの状態，すなわち仮の状態を示す「Tentative」，有効な状態を示す「Valid」，無効な状態を示す「Invalid」がある．端末が新しいアドレスを生成してから約 1 秒間は状態が「Tentative」の状態となり，この間に DAD 処理を行うことになっている．状態が「Tntative」のときは，パケットを受信することができない．

図 3 に DAD の処理シーケンスを MPPCv6 で移動後の通信を例にとって示す．MN はネットワークを移動するとルータから広告される RA に含まれるネットワークプレフィックスを用いて自らのアドレスを生成する．この時，生成されたアドレスの状態は「Tentative」である．アドレス生成後，CN に対してただちに CU Request を送信し，それを CN は受け取ることができる．CU Request を受けとった CN は CIT を更新して CU Response を返す．ここで CU Response をルータから MN に送信しようとしたとき，ルータには Neighbor Cache（NC）のエントリが存在しないため，エントリーを生成するために MN に対して NS（Neighbor Solicitaion）を送信する．ルータが持つ NC とは IPv4 の ARP キャシュ相当するもので，IP アドレスと MAC アドレスのマッピング情報が記録されている．この時，MN のアドレスの状態が「Valid」であれば NC を更新するための NA（Neighbor Advertisement）を送信可能であるが，状態が「Tentative」であるため NA を送信できない．ルータは MN からの NA が戻らないため NS の再送を行うが，状態が「Valid」となるまでで CU Response を送り出せない状態となる．この間の通信は断絶状態となる．また，DAD 処理が完了するまでに MN から NS を送信しているが，これは MN が生成したアドレスが重複していなかをチェックするパケットである．もし他の端末がそのアドレスを使用していた場合，MN 対して NA の応答が返される．

4.2.2 Optimistic DAD

DAD 処理時間の課題を解決するため Optimisitic DAD[7] と呼ばれる技術が存在する．Optimistic DAD とは，アドレスの状態として「Optimistic」状態を加え，NA パケットにも工夫を加えるものである．「Optimistic」状態のアドレスは限定的に「Valid」と同様に使用することが可能な状態で，DAD 処理終了後に「Valid」又は「Invalid」へ移行する．また，NA パケットにはオプションフィールドとして「Override」フラグ（O フラグ）が定義されている．O フラグが立っていればルータの NC エントリを強制的に上書きし，フラグが立っていなければ上書きをしない．通常の DAD 処理では，アドレスの状態が「Valid」の時にのみ NA を応答するため，必ず O フラグが立っている．既存のエントリがない場合はエントリを新規作成する．

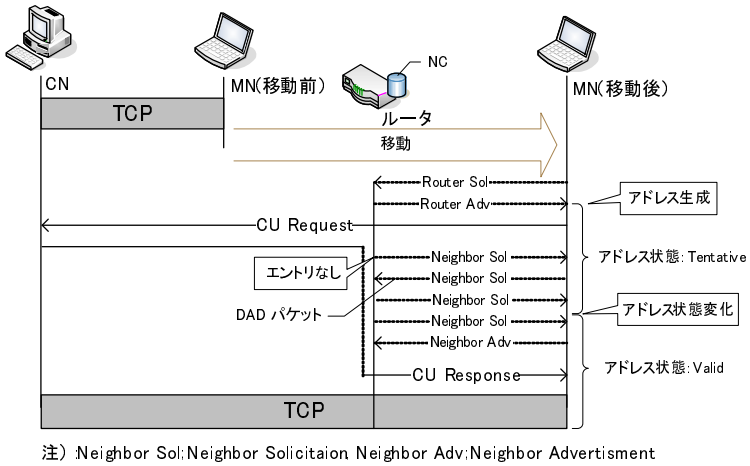

図 3 DAD の処理シーケンス
Fig. 3 Sequence of DAD Process.

Optimistic DAD ではアドレス生成後に状態が「Optimistic」となる．この状態時にはルータからの NS パケットを受信しても，NA パケットの O フラグを立てずに応答する．その NA を受信したルータは NA の送信元である MN に待機させてあるパケットを転送する．DAD 処理が完了するまではルータの NC エントリが作成されないので，ルータから MN へ 1 パケットを転送する度に必ずこの NS と NA のやり取りを行う．これにより，DAD 処理中でもルータの NC キャッシュを更新することなく通信を開始することができる．

しかし，この Optimistic DAD の課題は，ルータと端末の両方が対応していなければならない．多くの場合，ルータに Optimistic DAD が実装されていない．また，DAD の処理中は毎パケット毎に NS/NA のやり取りが発生するため，スループットが低下することが考えられる．

4.2.3 DAD 回避方式

MPPCv6 では，O フラグとアドレス状態を独自のタイミングで変更させることにより，

Vol.2010-MBL-52 No.5
2010/1/28

DAD 処理時間を短縮する．また，Optimistic DAD のような状態は追加せず，既存の NDP システムに与える影響を最小限にした．具体的には以下のような改良を行った．

(1) **CU Request 送信前に DAD のための NS パケットを独自に送信する．**

(2) **アドレスの生成後，状態を「Tentative」から「Valid」へただちに移行する．**

(3) **アドレス生成から 1 秒間，MN がルータに応答する NA の O フラグは立てずに送信する．**

図 4 に MPPCv6 に本方式を適用した場合の動作について示す．MN は移動後にルータから広告される RA を受信して，アドレスを生成する．この時，アドレスの状態は「Tentative」となっている．初めに，CU Requst を送信する前に DAD のための NS を送信する．この動作は (2) でアドレスの状態を「Valid」に変更させため，MN から DAD のための NS パケットが送信されない．NS を送信しないと MN はアドレスが重複したしたかどうかを検知できないため，MPPCv6 独自で送信する．次に，CU Request 送信前にアドレスの状態を「Valid」に変更する．この動作により MN はルータから送信される NS パケットに対し，DAD 処理時間を待つことなく NA を応答することができる．NA を応答することで CN から CU Response パケットがルータから解放されるため，迅速．また，(3) でルータに応答する NA の O フラグは立てないが，ルータに MN の NC エントリーが存在しないため新しいエントリーが作成される．このような処理を行うことにより，DAD の処理を大幅に短縮することが可能となった．

図 5 では MN が ON と重複したアドレスを生成した場合について示す．図 4 と同様に，MN はアドレス生成後，(1) と (2) の動作を行ってから CU Request を送信して，CN は応答として CU Response を返信する．ルータは CU Response パケットを受信後，NC のエントリーが存在しないため，NS を ON と MN へ送信する．NS を受け取った ON は通常の NA をルータへ返信する．同時に，MN もルータへ NA を返信するが，NA パケットの O フラグを立てずに返信する．ルータは二つの NA を受信するが，MN からの NA には O フラグが立っていないため，ON のエントリーが生成される．そのため，ルータからのパケットは必ず ON に届けられ，ON の通信を妨害することはない．また，CN からの CU Response も ON に転送されてしまうがことになるが，CU パケットは ICMPv6 をベースにしているため，ON の通信には影響を与えることがない．この場合，MN は独自で NS パケットを送信しているため，重複したことを知らせる NA が ON から返信される．そのパケットにより MN はアドレスの重複を知ることができる．MN は重複アドレスを検出すると，アドレスの再生成を行い，再度 CU ネゴシエーションを開始する．

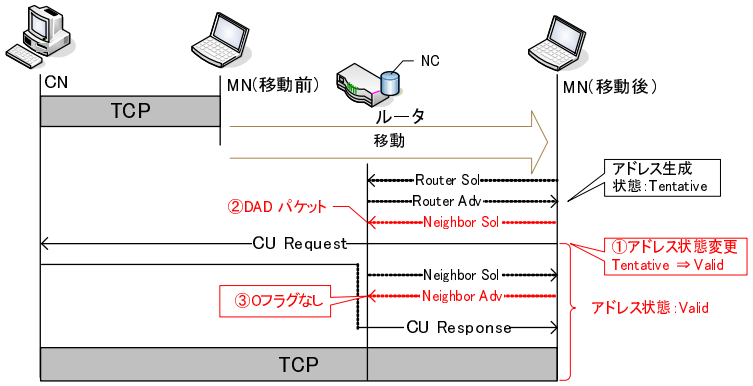

図 4 DAD 回避方式を用いた通信シーケンス
Fig. 4 The communication Sequence with the DAD avoidance method

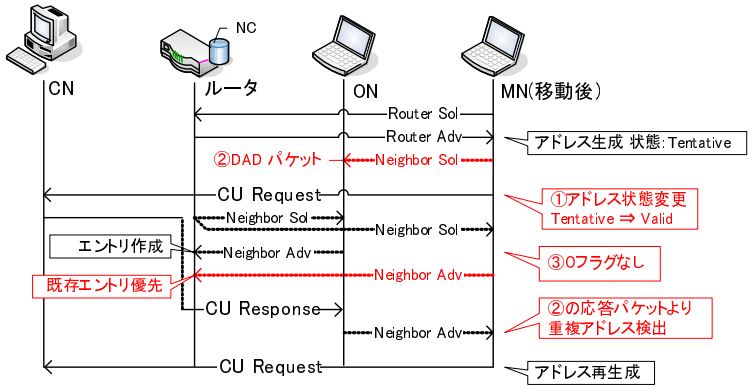

図 5 DAD 回避方式を用いた通信シーケンス（アドレス重複時）
Fig. 5 The communication Sequence with the DAD avoidance method.(when address duplicats)

Vol.2010-MBL-52 No.5
2010/1/28

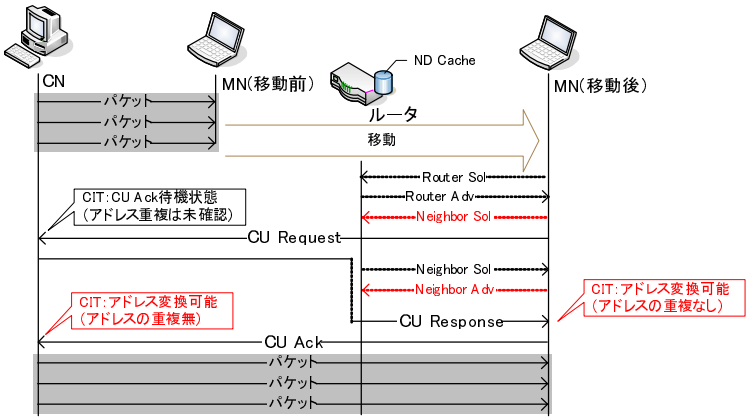

図 6 拡張した CU ネゴシエーション.
Fig. 6 The Extended CU negotiation.

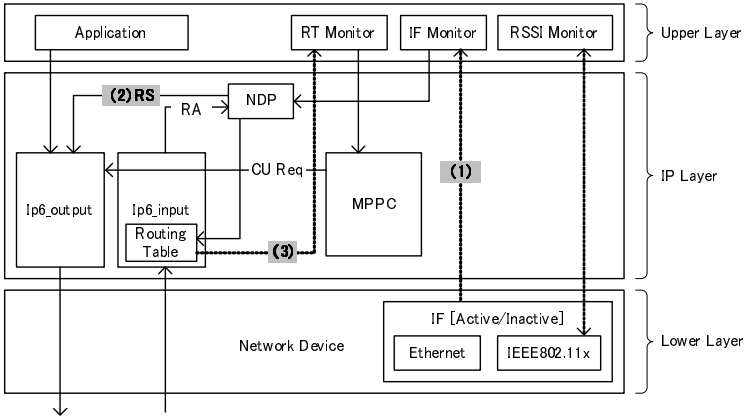

図 7 移動検知モジュールの実装.
Fig. 7 Implementation of move detection modules.

4.3 CU ネゴシエーションの拡張

DAD 回避方式の実装に伴い CU ネゴシエーションの拡張も行った．既存の MPPC では,CU Request を CN が受け取るとただちに CIT を更新してしまう．MN のアドレスが ON のアドレスと重複していた場合，MN 宛のパケットがすべて ON に届けられることになる．この状態で CN がパケットを送信すると，ON は予期しないパケットを受け取ることになり，ON の動作や通信を妨害する可能性がある．この課題を解決するために CU Ack メッセージを新たに定義して，CU ネゴシエーションを 3WAY に拡張する．

移動した MN が CU Request パケットを送信し，CN はそれを受信すると通常通りに CIT を更新が，この時 CIT の状態をアドレス変換させないように CU Ack 待機状態に遷移させる．次に，CN は CU Response パケットを MN に送信することで MN は CIT を更新する．MN の場合は従来通りアドレス変換を行える状態である．最後に，MN は CN に CU Ack パケットを送信し，それを受けたった CN は CIT をアドレス変換可能な状態に遷移する．もし，CU Ack が MN から応答されない場合は MN のアドレスが重複したと判断し，パケットを送信しない．以上のように CIT の状態遷移のタイミングをずらすことで，DAD 回避方式を用いても他の端末への影響はない．

4.4 移動検知機能

MIPv6 の移動検知方法はルータから一定の間隔で広告されている RA に依存しているため，移動を検知するまでの時間が RA の送信間隔に大きく左右される．そこで，MPPCv6 ではインターフェースがネットワークに接続されたかどうかを監視することにより，端末側から RA の要求を行えるように工夫した．**図 7** に移動検知システムモジュール図を示す．初めに (1) より，Ethernet であればコネクタが接続されているかどうか，IEEE802.11X であれば無線 AP に接続されているかどうかをアプリケーションで検出する．このインターフェースの監視を行うことで，移動したかどうかを判断するための RS を実行する (2)．この機能により移動検知のトリガとなる RA をただちに受信することができるため，定期的に広告される RA に頼る必要がない．もし，ネットワークを移動していれば新しいアドレスを生成し，ルーティングテーブルへ新しい IP アドレスが追加される．この処理を (3) のルーティングテーブル監視デーモンより取得し，カーネルの MPPC モジュールへ CU Request 送信要求を行う．このような実装を行うことで，ルータの RA 広告間隔に左右されることなく安定した移動検知が可能となった．

5. 性能評価

MN と CN に実装された MPPCv6 の性能を計測するために，ネットワークトラフィックの生成ツールである iperf を用いて TCP 通信を行った．実験環境を**図 8** に示す．IPv6 ルータ 3 台を用いて IPv6 ネットワークを構築し，R2 と R3 には無線 AP2 と AP3 をそれぞれ

Vol.2010-MBL-52 No.5
2010/1/28

表 1 装置仕様

Table 1 Device specification

	MN	CN
CPU	Core Duo U2500 1.2GHz	Pentium 4 3.0GHz
Memory	1014MB	512MB
NIC	Intel 3945ABG	100Base-TX
OS	FreeBSD 7.0	FreeBSD 7.0

接続する．無線の規格には 802.11a を用い，セキュリティや認証は行っていない．MN は無線接続を AP2 から AP3 へ切り替えることにより無線ハンドオーバ実験を行った．本実験に用いた各装置の仕様を**表 1** に示す．また，通信断絶から通信開始までに要する時間，すなわち通信断絶時間を図 4 のシーケンスから以下のように分ける．

(1) **L2 ハンドオーバ (通信断絶〜RS 送信)**
(2) **アドレス生成 (RS 送信〜NS 送信)**
(3) **CU ネゴシエーション (CU Req 送信〜CU Res 受信)**
(4) **通信再開 (CU Res 受信〜通信再開)**

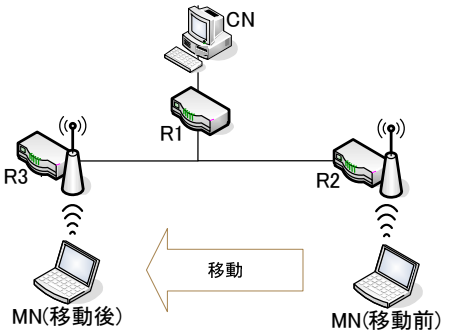

図 8 システム構成.
Fig. 8 System configuration.

5.1 ハンドオーバ処理時間

CN から MN に対して 10 秒間の TCP トラフィックを送信し，MN を R2 から R3 へ移動させた．この場合においての MN の通信断絶時における各ハンドオーバ処理時間を**表 2** に示す．MPPCv6 を用いて通信を継続した場合，ハンドオーバ処理時間の合計は約 3[s] という結果が得られた．(1) の処理は通信断絶中の処理の中でも全体の 3 分の 2 を占めており，約 2[s] の時間を要する．文献[8] では無線 AP の切替は 50〜400[ms] になると推測していたが，倍以上の時間がかかってしまった．(2) の処理では IPv4 の DHCP に要する時間に比べ，IPv6 のアドレスの取得時間は大幅に短くなっている．IPv6 のアドレス自動生成は標準機能であるため，DHCP のようなシーケンスがなく，相性による遅延もない．アドレスの取得部分に関しては予測通りの結果が得られている．また，MN が RS を送信してルータから RA の応答がされるまでの時間にばらつきがみられたこれは，ルータが RA を応答する間隔がある設定された時間の間からランダムに決定される仕様のためだと考えれる．(3) では MPPC 特有の処理全体を表しているが，他の処理と比べても十分短い処理時間であると言える．MPPC の各処理については次節で詳しく述べる．(4) の時間も平均では 0.7[s] となっているが，データを見るとばらつきがあった．これは今回の実験で TCP を用いたため，TCP の再送制御に依存してしまったが原因と考えられる．この課題への対策としては，CU ネゴシエーション中に再送されたパケットを MPPC の CIT 内部にバッファしておくことで解決できる．ネゴシエーション完了直後にバッファしたパケットを開放することで短時間での通信再開が可能となる．

また，MIPv6 の通信断絶時間は 4[s]〜5[s] ほどかかる．MIPv6 の RA の受信をトリガとした移動検知の手法からすると，断絶時間がこれ以上長くなることも十分考えられる．実験環境は有線通信ではあるが，現段階では MPPCv6 のほうが通信斬絶時間が短い．この結果の違いは移動検知機能と DAD 回避方式の実装に起因していると考えられる．

5.2 DAD 処理時間

表 3 に MPPC ネゴシエーション中に発生する DAD 処理時間について示す．4.2 で述べた DAD 回避方式を実装したものと未実装のものを全体の合計で比べると約 2[s] ほど短縮されているのがわかる．MPPC の処理自体が非常に短いため，ほとんどの処理時間を DAD が占めていた．DAD 処理の時間はおよそ 1 秒間であるが，アドレスの状態が「Valid」に変化してから受信した NS にの応答する．そのため，NS の受信タイミングの差で平均で 1.5[s] ほどの時間がかかる．この結果から，MPPCv6 において DAD 回避方式が有効だと確認ができた．また，CU Request を送信して後，ルータが送信する NS を受信するまで多

表 2 ハンドオーバ時における各処理時間

Table 2 Proccess time in the handover sequence

処理内容	処理時間 [単位]
(1) L2 ハンドオーバ	1.931[s]
(2) アドレス生成	0.276[s]
(3) CU ネゴシエーション (L3 ハンドオーバ)	96.5[ms]
(4) 通信再開	0.708[s]
合計	3.004[s]

表 3 MPPC ネゴシエーションと DAD 処理時間の詳細

Table 3 Details of DAD process time in MPPC.

処理内容	DAD 回避方式	未実装
CU Req 送信〜NS 受信	95.21[ms]	91.55[ms]
NS 受信〜NA 送信	0.303[ms]	1.906[s]
NA 送信〜CU Res 受信	0.443[ms]	0.756[ms]
合計 (CU Req 送信〜CU Res 受信)	95.79[ms]	2.005[s]

少時間がかかっている．CN で CU Request を受信してから CU Response を送信するまでは 2[ms] ほどの時間なので，ルータが DAD 処理を行っているためだと推測される．

6. ま と め

本稿では IPv4 で有効性が示されていた MPPC を IPv6 に対応させた MPPCv6 の実装について報告した．IPv6 における移動透過性の課題となる移動検知や DAD 処理に独自機能を実装することにより対応した．移動検知においてはインターフェースやルーティングテーブルの監視を行うことにより，迅速な移動検知が可能となった．DAD 処理については MPPC の特徴を生かして NA の O フラグを制御することにより処理時間が短縮された．

また，ネットワーク環境は IPv6 の普及に伴い，IPv4 と IPv6 ネットワークが混在すると考えられる．そのような IPv4/IPv6 混在ネットワーク環境でも対応可能な Mobile PPC の検討と実装を行っていきたい．

謝辞 本研究の一部は，日本学術振興会科学研究費補助金（特別研究員奨励費 20・1069）の助成を受けたものである．

参 考 文 献

1) 寺岡文男：インターネットにおけるノード移動透過性プロトコル，電子情報通信学会論文誌 (D-I)， Vol.J87-D1, No.3, pp.308–328 (2004).
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) Perkins, C.: IP Mobility Support for IPv4, RFC 3220, IETF (2002).
5) Kent, S.: Security Architecture for the Internet Protocol, RFC 4301, IETF (2005).
6) Vixie, P., Thomson, S., Rekhter, Y. and Bound, J.: Dynamic Updates in the Domain Name System (DNS UPDATE), RFC 2136, IETF (1997).
7) Moore, N.: Optimistic Duplicate Address Detection (DAD) for IPv6, RFC 4429, IETF (2006).
8) MISHRA A, S.M. and W, A.: An Empirical Analysis of the IEEE 802.11 MAC Layer Handoff Process, *ACM SIGCOMM Comput Commun Rev*, Vol.33, No.2, pp. 93–102 (2003).