目次

概要

無線ネットワーク環境の広がりにより, 多くのモバイル端末がどこからでもネットワークに接続できるようになってきている．この様な背景から，自由に移動しながらネットワークに接続したいというニーズが広まっている．しかし，インターネットに接続中の端末が移動するとIPアドレスが変化し, 通信が切断されてしまう．

このため通信中に移動しても，通信に影響を与えない移動透過性が要求される．これまで移動透過性を実現する技術はいくつか提案されているが, 多くの場合第３の装置による基盤が必要となる．このような装置の存在は，今後普及する P2P 通信の特徴を損なううえ，二重化等の措置をとるために，管理負荷が増すなどの課題がある．

そこで本稿では，モバイル端末の IP アドレスが変化した場合に，両エンド端末においてアドレス変換処理を実行する Mobile PPC (Mobile Peer to Peer Communication)を提案する．Mobile PPC は，上位ソフトウェアを一切変更する必要が無く，パケット長が変化しないため高スループットを実現できる．Mobile PPC を実装し評価をした結果, 高スループットを確保したままで移動透過通信が行えることを実証した．

第1章 はじめに

ノート PC や PDA などのモバイル端末を持ち歩き，行く先々でインターネットに接続して利用するユーザが増加している．この様な状況下では，通信中にユーザが移動しても，通信を継続できることが要求される．TCP/IP では，IP アドレスがノード識別子としての役割だけでなく位置の情報も含んでいるため，端末がネットワークを移動すると異なる 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として用いられており、IP アドレスが異なると別の通信と見なされ通信を継続することができない．この課題を解決するために，端末が移動しても通信を継続できる機能を移動透過性と呼び，これまで多くの方式が研究されている[1].

移動透過性の研究を大きく分類すると，特殊な中継サーバを用いるプロキシ方式とそれを必要としないエンドツーエンド方式がある．プロキシ方式は，移動ノードと通信相手ノードの間にプロキシサーバが介在し，プロキシサーバが移動ノードの IP アドレス変化を通信相手ノードから隠蔽する．エンドツーエンド方式はエンド端末間で課題を解決し，上位ソフトウェアに対して IP アドレスの変化を隠蔽する．また，別の分類方法として，移動透過性を実現するレイヤの違いにより，ネットワーク層で実現する方式とトランスポート層で実現する方式がある．トランスポート層では通信識別子の制御がやりやすいという利点があるが，TCP または UDP のどちらに適用するかによりその方式が異なる．これに対し，ネットワーク層での実現方法は，TCP/UDP のいずれにも対応できる点で有効である．

Mobile IP[2-6]は，プロキシ方式をネットワーク層で実現する．プロキシサーバとして移動ノード(Mobile Node ; 以下 MN)の位置を管理するホームエージェント(以下 HA)を導入し，通信相手ノード(Corresponded Node; 以下 CN)側から MN への通信パケットは HA が代理受信し，MN へトンネリング転送を行う．MN 側から CN への通信パケットは直接送信される．CN は通信相手が HA のように見えるため通信識別子が変化せず通信を継続できる．Mobile IP は IETF での十分な検討を経て確立された技術であるが，HA という特殊な装置が必要である他，通信経路に冗長が発生したり，トンネリング転送時に余分なヘッダが必要になるなどの問題点が指摘されている．

MSOCKS[8]は，プロキシ方式をトランスポート層で実現する．プロキシサーバとして，socks サーバを導入する．DNS サーバには，MN のホスト名に対して socks サーバの IP アドレスを登録する．CN は socks サーバが通信端末であると認識する．socks サーバは MN と socks サーバ間, socks サーバと CN 間で確立された異なる TCP コネクションを結合しなおすことにより通信を継続する．MSOCKS は，ヘッダオーバヘッドは発生しないが，両方向の通信とも socks サーバを経由するので冗長な

経路が発生する．

TCP-R[9]，TCP Migrate[10]，MMSP[11]はエンドツーエンド方式をトンランスポート層において実現する．TCP-R，TCP Migrate は，MN の IP アドレスが変化したときに，TCP オプションフィールドを用いて MN から CN に変更情報を通知し，エンド端末間で TCP コネクションを張り直す．この方式では，TCP 機能の拡張が必要であり，またアプリケーションも TCP に限定される．MMSP は，UDP を拡張し，MN の IP アドレスが変化したときに，独自に定義したパケットで相手に通知する．IP アドレスの通知が完了するまでの間，新旧の IP アドレスを保持しておくことなどによりパケットロスを軽減する工夫をしている．この方式では，アプリケーションが MMSP に対応している必要があり，かつ UDP に限定される．

LIN6[12], MAT[13]はエンドツーエンド方式をネットワーク層において実現する．LIN6 では，IPv6 アドレス空間の内容をノード識別子と位置指示子という 2 種類の空間に分離させ，ノード識別子と IP アドレスの対応保持する位置管理サーバを設けることにより，IP アドレスの変化を上位ソフトウェアから隠蔽する．しかし，LIN6 では，アドレス空間のビット数が半分になることからアドレス利用効率が大きく低下する上，独自のアドレス体系をグローバルユニークに割り当てる必要がある．また，IPv4 ではアドレス空間が不足するため適用できない．MAT[13]は，LIN6 のこのような課題を解決するもので，アドレス空間を分割することはせず，ノード識別子と位置指示子に対応する IP アドレスを別途定義して両者を変換する．この方式では通常の IP アドレス体系を適用することができ，IPv4 でも同様の考えを適用できる．しかし，独自の位置管理サーバが必要になる点は変わっていない．また，MAT 非対応のノードは，通信開始時に MN がホームネットワーク上にいないと MN の位置指示子となる IP アドレスを知ることができず通信を開始することができないという課題がある．

Mobile IPv6[6]は，Mobile IP を IPv6 用に改造したもので，MN が移動後に経路最適化と呼ぶ機能が標準で追加され，冗長な経路を通さない通信が可能となった．しかし，通信開始時には HA を経由しなければならないため，HA が必須となることに変わりない．また，経路最適化時にはヘッダのオーバヘッドが常時発生する．

今後のユビキタス社会を想定するとネットワークを最大限に活かせる P2P (Peer-to-Peer)通信の要求がますます増加すると考えられる．プロキシ方式のように一般通信において特殊な装置を経由する方式では，P2P 通信の特徴である柔軟性やリアルタイム性が失われる懸念がある．また，エンドツーエンド方式でも特殊な位置管理サーバを必要とする方式は，十分な普及に至るまでその機能が発揮できないうえ，サーバの 2 重化などの対策が必須であり管理負荷が大きい． P2P 通信が個人間の通信が主体となることを踏まえると，エンドツーエンド方式でかつ，特殊な位置管理サーバを必要せずに移動透過な通信を実現できることが望まれる．また，実装レイヤについては，TCP/UDP の区別無く利用可能なネットワーク層での実現

方法が有利と考えられる．

本論文では，エンド端末の IP 層にアドレス変換処理機能を導入し，エンドツーエンド方式をネットワーク層で実現する Mobile PPC (Mobile Peer to Peer Communication)を提案する．Mobile PPC では，MN の IP アドレスが変化した時，MN から CN に対して直接変化情報を報告し，両端末の IP 層の中にアドレス変換テーブルを生成する．以後の通信パケットは上記アドレス変換テーブルに基づきアドレス変換する．この方式により，IP アドレスの変化は上位ソフトウェアから隠蔽することができ，移動透過性を容易に実現することができる．Mobile PPC を FreeBSD 上に実装し，動作確認と性能測定を実施した結果，Mobile PPC はスループットの低下がほとんどない移動透過性を実現できることを確認した．

以下，2 章で従来技術の例として Mobile IP，LIN6，MAT について記述し 3 章で Mobile PPC の原理と詳細について記述する．4 章で Mobile PPC の実装，5 章で性能測定結果と Mobile PPC の評価，6 章でむすびについて述べる．

第2章　従来研究

既存研究として，プロキシ方式の代表 Mobile IP，エンドツーエンド方式の代表 LIN6，MAT をとりあげる．いずれもネットワーク層による実現方法であり，トランスポート層より上位のソフトウェアに一切影響を与えないという利点がある．

第2.1節 Mobile IP

図 2-1 に Mobile IP の通信を示す．MN は移動によって変化しないホームアドレス(HoA)と，移動先ネットワークで割り当てられる気付アドレス(CoA)の二つの IP アドレスを持つ．HA は，MN の HoA と CoA の対応付けを行い，HoA 宛のパケットを代理受信し，CoA 宛に転送する役割を持つ．

Mobile IP の動作は，HA への登録，データ通信に分けることができる．MN は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新しく取得した CoA を HA へ登録する．HA は MN の HoA と CoA の対応付けを更新する．CN から MN へ通信パケットを送信する場合は，宛先を HoA とする．HA はこのパケットを代理受信し，CoA 宛の IP ヘッダでカプセル化して MN に転送する．MN から CN への通信パケットは CN 宛に直接送信される．このとき送信元アドレスは HoA とする．

Mobile IP は，このように HA という特殊な装置を導入し，CN が常に 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実現する．MN 宛のパケットは必ず HA を経由するため，通信経路が冗長な三角経路となり，HA と MN 間は IP トンネルとなる．また，MN から CN へパケットを送信する場合に，送信元アドレスとして使われる HoA は MN のインターネット上での位置を正しく表していないため，途中のルータが送信元アドレスを偽っている不正パケットと見なし，破棄する可能性がある．

Mobile IP は，クライアントサーバ環境においては，CN として従来の固定サーバをそのまま利用できる点で有効である．しかし，P2P 通信が主体となる今後のネットワーク環境においては，必ずしも最適な方式とは言えない．

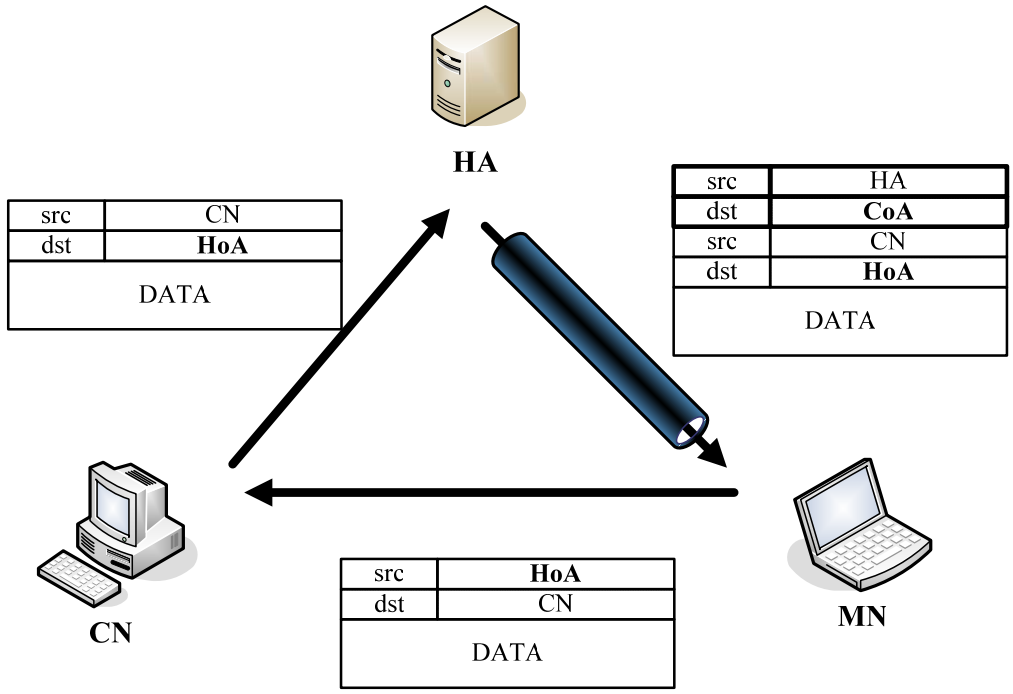

図 2-1　Mobile IP の通信

Fig.2-1　A communication method of Mobile IP.

第2.2節 LIN6

LIN6 (Location Independent Networking for IPv6)は，IP アドレスに含まれているノード識別子と位置指示子としての情報を明確に分離させ, IPv6 アドレス体系自体を見直す提案である．即ち IPv6 アドレスの上位 64 ビットを位置指示子，下位 64 ビットをノード識別子として扱う．また，上位 64 ビットに対し LIN6 プレフィックスと呼ばれる固定値を定義しておき，IP 層よりも上位層ではノード識別子と LIN6 プレフィックスを合わせた LIN6 汎用アドレス，下位層では位置指示子とノード識別子を合わせた LIN6 アドレスとなるように IP 層で変換を行う. 上位層ではノードの位置や移動にかかわらず常に LIN6 汎用アドレスを用いる.

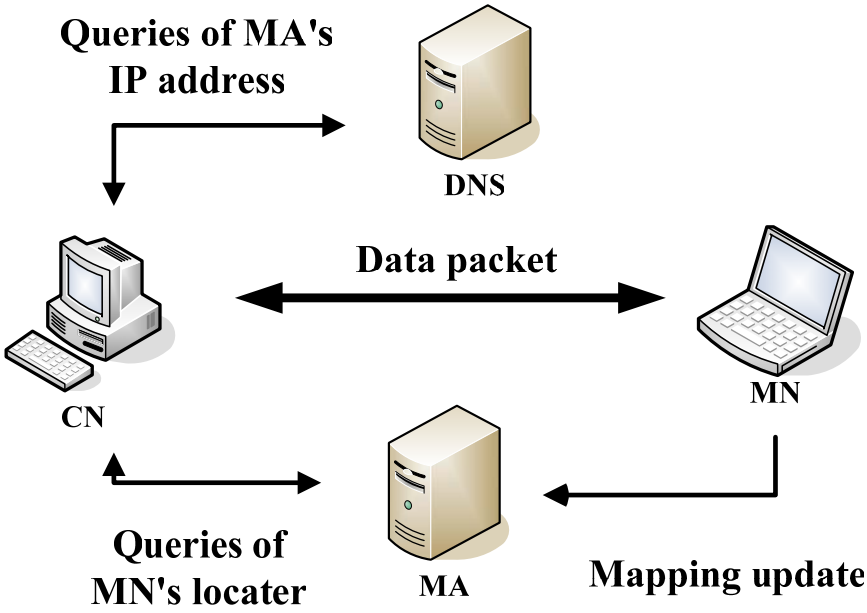

図 2-2　LIN6 の通信方式

Fig.2-2　A communication method of LIN6.

図 2-2 に LIN6 の通信方式を示す．LIN6 はエンドツーエンド方式であるため両端末は対等の関係にあるが，説明のため移動する側の端末を MN，通信相手側の端末を CN と呼ぶ．MA は MN のノード識別子と現在の位置情報との対応関係を常時保持している．CN が MN の LIN6 アドレスを知るためには，DNS からまず MA の IP アドレスを知り，MA から MN の LIN6 アドレスを取得する．MN が CN の LIN6 アドレスを知るときも同様の手順をとる．MN が CN と通信中に別のネットワークに移動した際には MA に位置指示子に変化を通知し，CN に対して MA から MN の LIN6 アドレスを再取得するように通知する．

LIN6 は，上記のように IP アドレスの役割を明確に分割したという点で評価できるが, IPv6 のアドレス構造を 2 分割するためアドレスの利用効率が大きく低下する．さらに，独自のアドレス体系を持つことになるため，ノード識別子のグローバルユニークな割り当てが必要となりその管理機構が必要になる．また，アドレス管理装置として MA のような特殊な装置が必要になる．IPv4 に対してはアドレス空間を分割する余裕がないため適用が困難である．

第2.3節 MAT

MAT (Mobile IP with Address Translation)は，LIN6 と同様にノード識別子（ホームアドレス）と位置指示子（モバイルアドレス）を示す２つの IP アドレスを定義しているが，両者の対応関係（以下，マッピング情報）を保持する位置管理エージェント IMS（IP Address Mapping Server）をネットワーク上に設置し，両者の間でアド

レス変換を行う点が異なる．

MAT も LIN6 と同様に DNS から IMS のアドレスを取得する．通信相手の IP アドレスを知る順序は，LIN6 における MA を IMS に置き換えたものと似ている．ただし，MN から CN へ通信を開始する場合には，新規に定義した IP ヘッダオプションを用いて, MN のホームアドレスを通知する. CN がパケットを返信する際には，通知されたホームアドレスを元に MA から MN のホームアドレスとモバイルアドレスの対応を取得する．MN が CN と通信中に別のネットワークに移動した際には IMS にモバイルアドレスの更新を通知する一方, CN に対して IMS から MN のマッピング情報を再取得するように通知する．

MAT では，ホームアドレスとモバイルアドレスはともに通常の IP アドレス体系を使用することができ，原理的に IPv4 と IPv6 のどちらにも適応することが可能である．

このように，MAT では LIN6 の考えをもとにしているが，アドレス変換を行うことで LIN6 の課題をいくつか解決している．しかし，マッピング情報を保持する特殊な装置が必要である点は同様である．また，DNS に独自のレコードを追加するため MAT 非対応のノードは, 移動ノードのモバイルアドレスを知ることができず, 移動ノードに対して通信を開始することができないという課題が残されている．

第3章　Mobile PPC の提案

第3.1節 位置づけ

移動透過性を実現するためには，通信開始時において相手の IP アドレスを知る方法と通信中に IP アドレスが変わった場合に変更後の IP アドレスを知る方法の 2 つの機能を実現する必要がある．2 章で述べた技術では，いずれも上記 2 つの機能を 1 種類のアドレス管理装置(HA，MA，IMS)で実現しようと試みている．

Mobile PPC では両者の機能を明確に分け，前者を初期 IP アドレスの解決，後者を継続 IP アドレスの解決と呼ぶ．初期 IP アドレスの解決には，ホスト名と IP アドレスの関係を動的に管理するダイナミック DNS (以下 DDNS)[14]を用いることができる．DDNS は DNS の延長技術であり，既に実用になっている．MN は初期立ち上げ時や移動時に新たな IP アドレスを取得すると必ず DDNS にその情報を登録するため，初期 IP アドレスの解決が可能である．以後の説明では，通信が開始されるときには DDNS により初期 IP アドレスの解決が行われていることを前提とする．本論文における提案 Mobile PPC (Mobile Peer to Peer Communication)は継続 IP アドレスを解決するための技術と位置づけられる．

第3.2節 概要

Mobile PPC の機能は，移動情報の通知処理と IP アドレスの変換処理に分けられる．通知処理は，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新しく取得した IP アドレスを CN に通知する．通知処理により，MN と CN は移動前と移動後の IP アドレスの対応関係を記すテーブル（Connection ID Table ; 以下 CIT）を更新する．CIT レコードは，通信が開始される際にコネクション単位で生成されるもので，MN が移動するたびにその内容が書き換えられる．IP アドレスの変換処理は, 全てのパケットに対して CIT を参照しながらアドレス変換を実行する．このような方式により，上位層に対しては移動による IP アドレスの変化が隠蔽され，上位層はアドレスの変化に気づくことなくコネクションを維持できる．

Mobile PPC は，通常の IP アドレス体系をそのまま使用できる．エンドツーエンド方式によるアプローチであり，経路の冗長は発生しない．また，特殊な装置が不要である．IP アドレスの変換は，IP 層で行われるため，上位ソフトウェアを一切変更する必要が無い．また, IP アドレスを単に変換する方式であるためパケット長が変化することがなく高スループットが期待できる．Mobile PPC 機能を保持しな

い一般端末とは上位互換性があり，Mobile PPC を実装している装置と実装していない端末間であっても，移動しない限り通信は可能である．なお，本論文では IPv4 での実現を試みたが，原理的に IPv6 でも同様の考え方を適用可能である．

第3.3節 移動通知処理

図 3-1 に Mobile PPC による移動情報の通知方法を示す. MN と CN 間で新しく通信が開始されると, エンド端末は送受信パケットを元に上位層が認識する通信識別子情報が記された CIT (Connection ID Table)レコードを生成する．CIT レコードは，移動前と移動後の通信識別子の情報から構成される．通信開始時点では IP 層でのアドレス変換は実行されないので移動後の情報を示すフィールドには null が入っている．

MN が CN と通信中に別のネットワークへ移動すると, MN は移動先で DHCP[15]サーバなどから新しく IP アドレスを取得する．ここで MN は，新 IP アドレスとコネクション識別子の情報を含む CIT UPDATE (以下，CU)パケットを生成し，CN に宛てて送信する．CU は CN に対して移動を通知するとともに CIT の更新を要求する．CN は，通知された情報を元に自身の CIT を更新し，CIT の更新が完了したことを通知する CU 応答パケットを返信する．MN は，CU 応答パケットを受信後に自身の保持する CIT を更新する．エンド端末で更新された CIT は，MN の移動前と移動後の IP アドレスの対応関係が登録され，以後の通信パケットに対する IP アドレス変換処理に用いられる．

CU および CU 応答は ICMP Echo Request をベースに定義されている．CU のフォーマットを図 3-2 に示す．CU と CU 応答は共通のフォーマットである．ヘッダ部には CU/CU 応答の識別(type)，MN と CN 間の通信数(connection count)，CU の識別番号(CIT Update ID)，データ部には，新 IP アドレス(New IP address)と初期コネクション識別子(Initial connection identifier)が通信数の分だけ含まれる．初期コネクションとはエンド端末のトランスポート層が通信を識別する際に用いるコネクション識別子である．

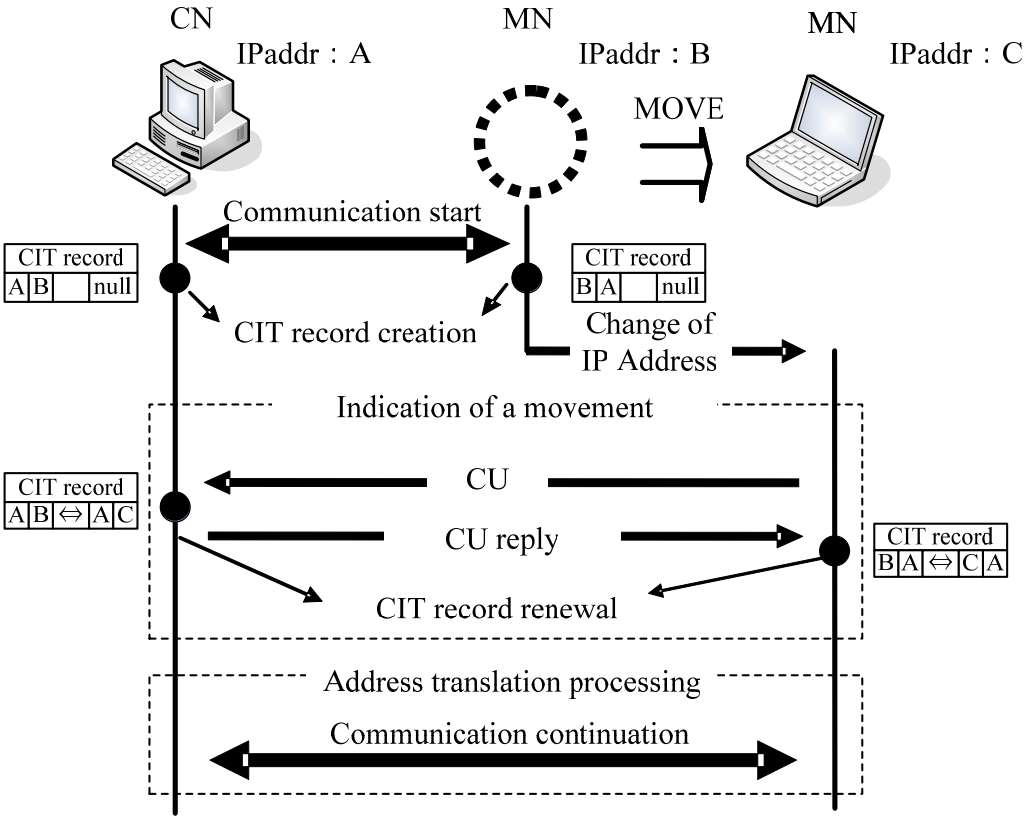

図 3-1　移動情報の通知

Fig.3-1　The notice of move information

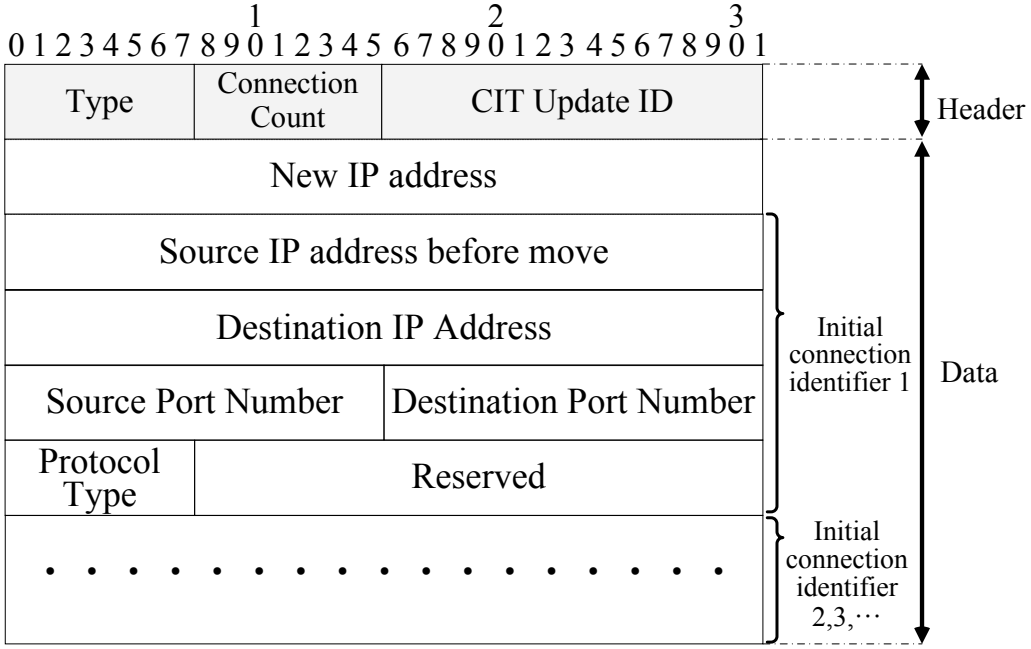

図 3-2　CU フォーマット

Fig.3-2 CU packet format

第3.4節 アドレス変換処理

図 3-3 に MN の IP アドレスが B から C へと変化した場合の IP アドレス変換処理を示す．CN から送信されるパケットの宛先 IP アドレスは，IP 層で CIT の情報を参照し移動後の IP アドレス C へ変換される．このパケットを受信した MN は，同様に CIT を参照しパケットの宛先 IP アドレスを移動前の IP アドレス B へ変換を行い上位層へ渡す．逆方向のパケットについても上記と同様なアドレス変換を行う．

このように IP 層において正しくルーティングされるようにアドレス変換し，上位層に対してはその変化を隠蔽するため通信中に MN が移動してもコネクションを維持させることが可能となる．

図 3-4 に IP アドレス変換の適用範囲を示す．図 3-4 は MN と CN が通信中に MN が移動を繰り返し，IP アドレスが B から C，C から D へと変化した場合を表している．IP アドレス変換処理は，通信開始時点における MN の IP アドレスがベースとなる．移動を繰り返した場合には，通信開始時の IP アドレスと移動先で取得した IP アドレスとの間で IP アドレス変換を行う．移動後に，新しく別の通信が開始された場合には，その時点における IP アドレスで通信が開始される．このように Mobile PPC によるアドレス変換時には，通信毎に上位層で認識する自分の IP アドレスが異なる状態となる．

図 3-4 に示す例では，MN が移動を繰り返し IP アドレス D が割り当てられているが，移動前の IP アドレス B および C はともに，上位層で通信を識別する目的に利用される．また，これらの IP アドレスは，実際に MN が保持する IP アドレスではないため，別の移動端末により割り当てられることも可能である．まれなケースとして，アドレス変換を適応する通信と，新しく開始する通信の上位層における通信識別子（両端末の IP アドレス，ポート番号，プロトコル番号）が一致するする可能性が考えられる．この場合，Mobile PPC では通信識別子の一致を検出すると，IP アドレス変換に加えポート変換を適応する．この方法により通信識別子の一致を必ず防止することができる．従って，CIT 内の変換情報として IP アドレスだけでなくポート番号も含まれている．

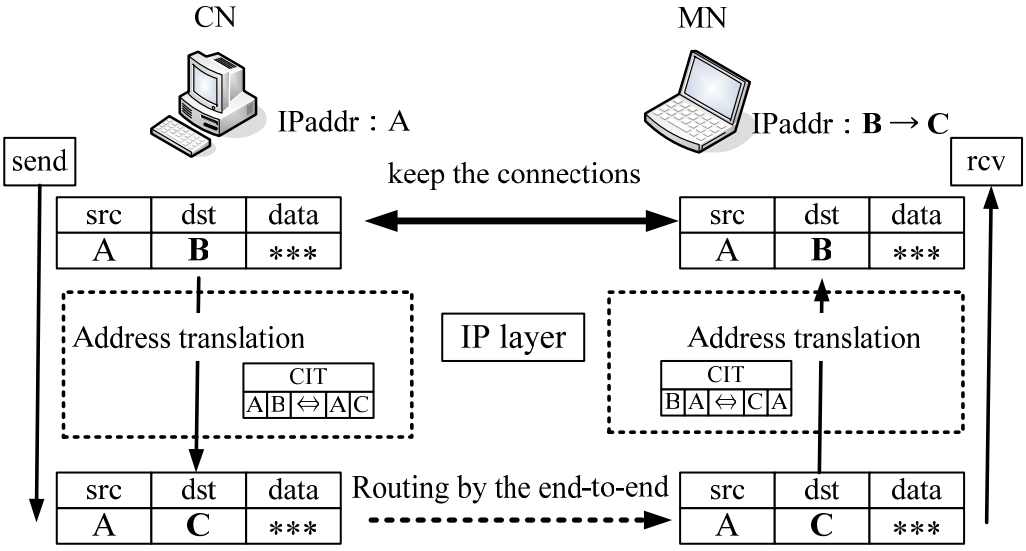

図 3-3　IP アドレス変換処理

Fig.3-3　Processing of address translation

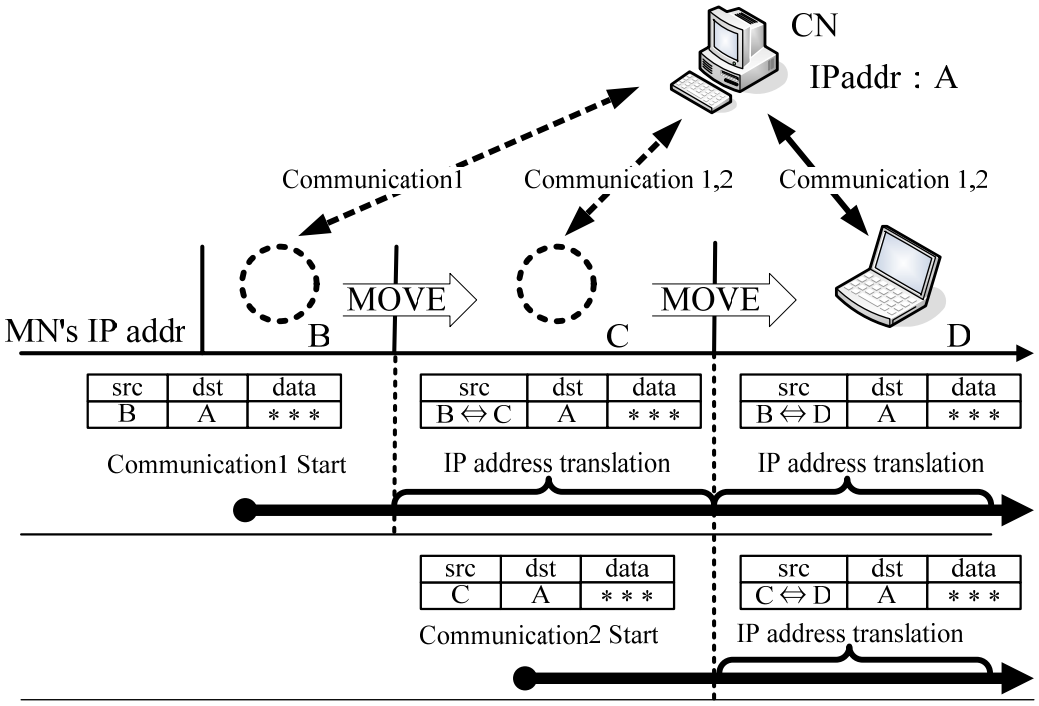

図 3-4　IP アドレス変換の適応範囲

Fig.3-4 Range of adjustment of address translation

第4章　Mobile PPC の実装

Mobile PPC を FreeBSD5.2.1 上に実装し動作を検証した．本章では Mobile PPC のモジュール構成と CIT のフォーマット詳細について記述する．

第4.1節 モジュール構成

Mobile PPC におけるモジュール構成を図 4-1 に示す．パケット受信時には IP 入力関数である ip_input から，パケット送信時には IP 出力関数である ip_output から Mobile PPC ソフトウェアを呼び出し，アドレス変換処理を終えたら差し戻す形をとっている．IP アドレス変更時には ARP 関数より Mobile PPC ソフトウェアが呼ばれ，移動通知処理を行う．IP 層以外の部分には一切変更を加えない．

Mobile PPC を実現するモジュールは CIT 操作モジュール，IP アドレス変換モジュール，移動管理モジュールの 3 つがある．

CIT 管理モジュールは，IP アドレス変換・移動通知モジュールから呼ばれ，CIT レコードの検索・生成・更新を実行する．また CIT を監視し，無通信状態の CIT レコードを削除する．

アドレス変換モジュールは，パケットの送信および受信時に呼び出される．入出力パケットのコネクション識別子をキーとして CIT 検索を行い，IP アドレス変換処理が必要であれば，CIT レコードの内容にしたがって，IP アドレスを変換し，それにともなうチェックサムの差分計算を行う．

移動管理モジュールは，IP アドレス変更時における CU および CU 応答パケットによる移動通知処理を行う．MN がネットワークの移動を行った場合，移動先で DHCP による IP アドレスの取得を行う．しかし，DHCP サーバから IP アドレスの使用許可 DHCP ACK を受信した時点では，IP アドレスがまだ確定しておらず，その後に必ず Gratuitous ARP を用いた IP アドレスの重複チェックが行われる．そのため，移動管理モジュールは，上記 ARP の重複チェックタイムアウトと同時に呼び出される．

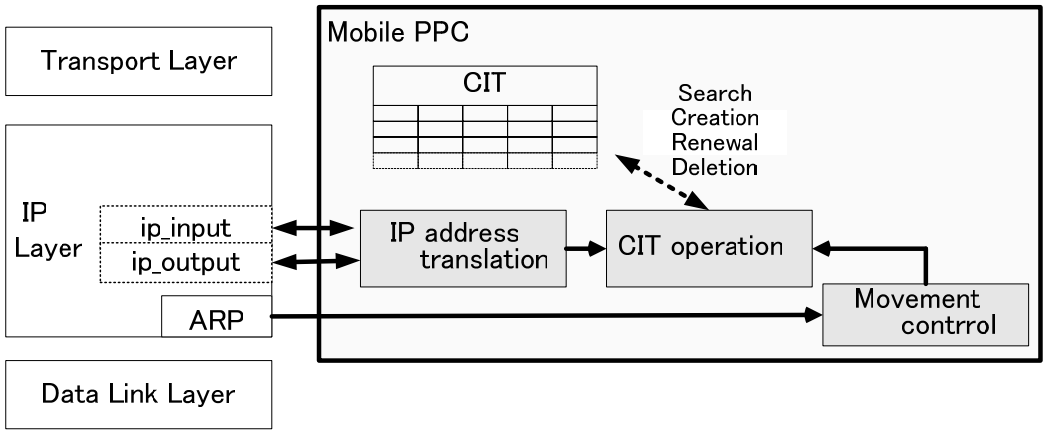

図 4-1　モジュール構成

Fig.4-1　Processing in IP layer

第4.2節 CIT

CIT のフォーマットは図 4-2 のとおりである．CIT は，通信開始時の初期コネクション識別子，新アドレス情報，変換処理フラグ，無通信カウンタ(cnt)，次テーブルアドレス(next)からなり，2048 レコードから構成される．ここで，初期コネクション識別子はトランスポート層が通信を識別するために持つ情報で、宛先/送信元の IP アドレス(sIP/dIP)，ポート番号の組(sport/dport)，プロトコル番号(proto)の 5 つの情報から成る．初期コネクション識別子は，3.2 節で述べたように通信が開始された際に登録される．新アドレス情報は，移動後の IP アドレスの組(tsIP/tdIP)とポート番号の組(tsport/tdport)からなり, CU および CU 応答による通知処理によって登録されるフィールドである．通常はポート変換を行わないが，3.3 節で示したように MN の移動前のアドレスが他のノードに割り当てられた場合, 上位層で認識する初期コネクション識別子が常にユニークになるように, 必要な場合に限りポート変換を適応する．

変換処理フラグ(trans)は，該当する通信パケットに対してアドレス変換が必要かどうかを示す．通信開始時は OFF でありアドレス変換を行わない．MN が移動して，新アドレス情報が登録された時に ON となり，以後の通信パケットにアドレス変換が適応される．無通信カウンタ(cnt)は，該当する CIT レコードを削除するためのフィールドである．この値は，スケジューラにより定期に的にデクリメントされるが，テーブルが検索されるごとに初期値（n）に戻される．値が 0 になると該当する端末間の通信が行われていないと判断され，該当レコードは削除される．CIT はハッシュテーブルとして実装し, 検索キーは送受信パケットのコネクション識別子のハッシュ値である．テーブルアドレス(next)は，ハッシュ値が衝突した場合に次のリンク先のテーブルアドレス(NAD)を示す．1 つのコネクションに対して送信用と受信用の 2 つの CIT レコードが生成される．CIT が更新された場合には，旧レコードは削除される．

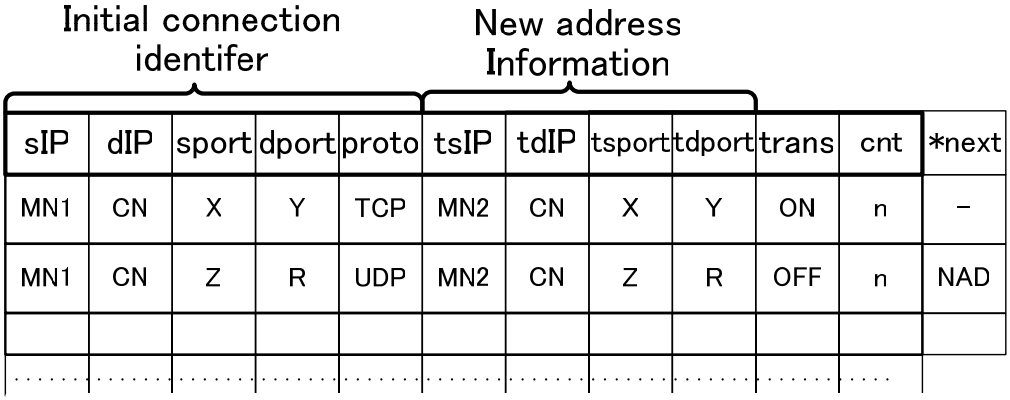

図 **4-2 CIT** のフォーマット

Fig.4-2 CIT Format

第5章　Mobile PPC の性能測定と評価

Mobile PPC を試作し，両エンド端末が移動を繰り返しても通信を継続できることを確認した．本章では，試作システムの性能測定結果，および同一条件下における Mobile IP とのスループット比較を行った．また，他の既存技術との比較を行った

第5.1節 パケット処理時間

Mobile PPC モジュールのパケット処理時間の測定結果を表 5-1 に示す．ここでパケット処理時間とは，ip_input/ip_output から Mobile PPC モジュールが呼び出され Mobile PPC による IP アドレス変換処理が行われた後，差し戻すまでの時間である．これは Mobile PPC を実装することにより，全てのパケットに対して CIT 検索が行われるため，これらにかかるオーバヘッドを調査するものである．Pentium (1.8GHz) の CPU を搭載し，100BASE-TX で接続された PC 上で RDTSC (ReaD Time stanp Counter)を用いて測定した．測定結果は FTP の通信中に流れた 1500 バイトの通信パケット 1000 個の処理時間の平均である．測定結果は，アドレス変換を行わない場合は 0.31 μ 秒，アドレス変換を行う場合は 0.54 μ 秒であった．1 パケット中継にかかる全体の処理時間は約 21 μ 秒であり，パケット処理時間の占める割合はアドレス変換を行わない場合は 1.47%，アドレス変換を行う場合は 2.53%であった．このことから Mobile PPC を実装したことによるオーバヘッドの増加は十分小さいと言える．

表 5-1　パケット処理時間

Table.5-1 Packet processing time

アドレス変換の有無	変換なし	変換あり
パケット処理時間の平均 [μ 秒]	0.31	0.54
(パケット処理時間の占める割合[%])	(1.47%)	(2.53%)
パケット中継にかかる処理時間 [μ 秒]	21.03	21.26

第5.2節 移動時間の測定

Mobile PPCの移動透過性にかかわる処理時間を図5-2に示す測定環境で測定した．2つのルータによりサブネットが異なる3つのネットワークを用意し，MNの移動先となるネットワークにはDHCPサーバを設置した．実験で用いた装置のOSは全てFreeBSD (5.2.1-R)であり，MNとCNにMobile PPCを実装している．DHCPサーバおよびクライアントは，ISC (Internet Systems Consortium)のISC DHCP v2 パッケージを使用し，パラメータはプロトコルデフォルト値を使用した．有線 LAN は100BASE/TX で構成し，MNはIEEE802.11bで接続した．MNからCNへ連続的にFTP を用いたデータ転送を実行させておき，MN を別のネットワークに移動させ，MN 側で直接コマンドに入力することにより，DHCP サーバから新しく IP アドレスを取得させた．

MNが異なるネットワークへ移動した際に発生するシーケンスは図5-3のとおりであり，通信を再開するまでの間，通信中断時間が発生する．通信中断時間はMNがDHCPサーバからのIPアドレス取得時間とエンド端末間で行われる移動通知処理時間の合計である．IPアドレス取得時間については，本提案方式の主題ではないが参考のために測定を行った．

表5-4にIPアドレス取得時間の測定結果を示す．この時間にはMNとDHCPサーバ間の2往復のDHCPシーケンスとIPアドレス取得後に行われるARPによる重複アドレスチェックが含まれる．表5-4に示すように約2〜5秒の時間を要し，通信中断時間のほとんどの割合を占める．表5-5に移動通知処理時間の測定結果を示す．移動通知処理時間には，MNとCNのCIT更新時間，CUパケットおよびCU応答パケットの伝達時間が含まれる．表5-5より移動通知処理時間は，パケットの伝達時間が大半をしめていることがわかる．パケット到達時間に多少ゆらぎがあるのは，測定環境に無線LANがあり，周囲の環境による影響を受けたためだと考えられる．

エンド端末間で行われているコネクション本数を1から5に増やした場合，MNとCNのCIT更新時間は0.038から0.047ミリ秒となった．このことから，エンド端末間のコネクション数による影響はほとんどないと言える．表5-4，表5-5からわかるようにMobile PPCによる移動通知処理時間は合計0.3ミリ秒程度であり，ほとんど無視できる．それに対して，DHCPサーバからのIPアドレス取得時間は平均3.34秒を要している．即ち，通信中断時間を減少するにはMobile PPCでなく，IPアドレス取得時間を短縮することが重要であることがわかる．本件の解決策としては，DHCPソフトウェアの最適化によるシームレスハンドオーバ[16]や重複アドレスチェックの高速化[17]などが有効と考えられる．

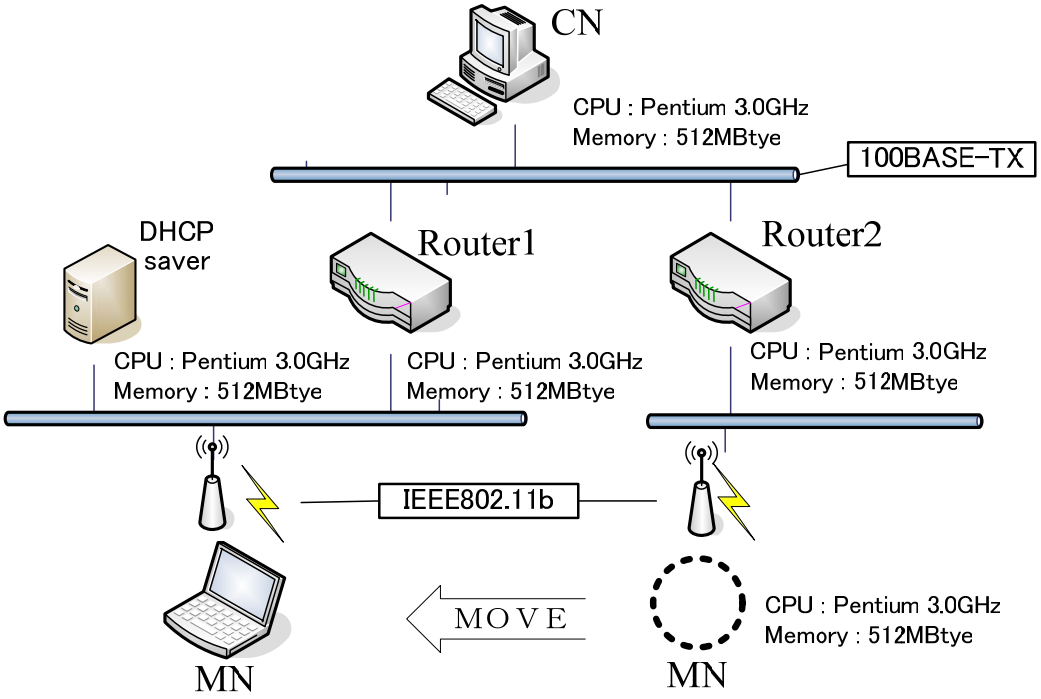

図 5-2 測定環境

Fig.5-2 Measurement environment

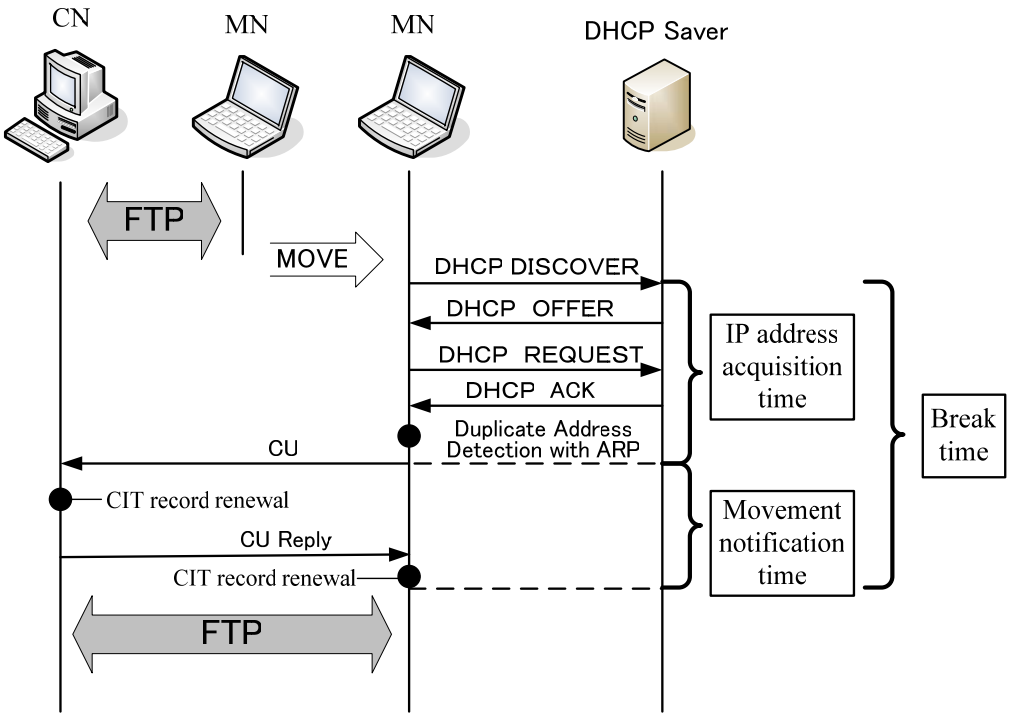

図 5-3 移動時のシーケンス

Fig.5-3 Sequence when moving

表 5-4 移動時のシーケンス

Table.5-4 IP address acquisition time by DHCP

	IP アドレス取得時間
最大	4.85 [秒]
平均	3.34 [秒]
最小	2.29 [秒]

表 5-5 移動通知処理時間

Table.5-5 Movement notification time

	エンド端末間の通信数				
	1	2	3	4	5
MN/CN の CIT 更新時間の和[ﾐﾘ秒]	0.038	0.040	0.044	0.045	0.047
CU/CU Reply の 到達時間の和[ﾐﾘ秒]	0.288	0.258	0.253	0.267	0.326
移動通知処理時間[ﾐﾘ秒]	0.326	0.298	0.297	0.312	0.373

第5.3節 Mobile IP とのスループット比較

Mobile PPC と Mobile IP のスループット比較を行うために図 5-6 のような評価システムを構築した. Mobile PPC 環境(図 5-6-a)では MN,CN に Mobile PPC を実装し，Mobile IP 環境(図 5-6-b)では，MN に Mobile IPv4 を実装し，Router2 に HA の機能を実装した．Mobile IP には PSU (Portland State University)配布の Mobile IPv4 パッケージ(PSU Mobile-IP　release for FreeBSD 5.2.1)を使用し，動作モードは Mobile PPC との比較をしやすくするため FA が不要な co-located care-of address モードとした．

実験で用いた装置仕様を表 5-7 に示す．無線は周囲の条件に依存するため，評価システム内は全て 100base-TX の有線 LAN とし，ネットワークの移動は，MN の LAN ケーブルを移動先ネットワークにつなぎ直すことでエミュレートした．図 5-6 中の矢印は，点線が MN から CN への通信経路，実線が CN から MN への通信経路を示している．

上記環境下で，以下のスループットを測定した．case1; Mobile PPC を実装しない状態でのスループット，case2; MN と CN が Mobile PPC を実装しているがアドレス変換をする前のスループット（移動前），case3; Mobile PPC を実装し，かつアドレス変換をしている時のスループット（移動後），case4;MN が Mobile IP を実装し HA

配下にいる時のスループット（移動前），case5;Mobile IP を実装して IP トンネリング通信をしている時のスループット(移動後)

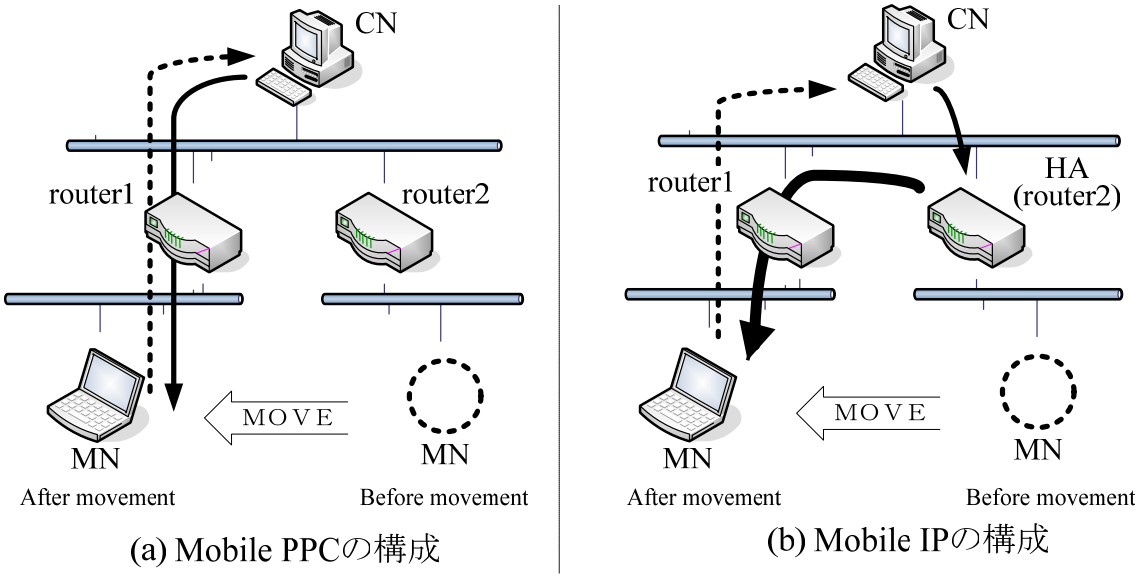

図 5-6 評価システムの構成

Fig.5-6 Structure of evaluation system

表 5-7 装置仕様

Table.5-7 Device specification

	MN / CN / router1 / router2
CPU	Pentium4 3.0GHz
Memory	512MB
NIC	100BASE-TX
OS	FreeBSD 5.2.1-RELEASE

表 5-8 に測定結果を示す．スループットの測定には，ネットワークベンチマークソフト Netperf[18]を使用し，20 回の平均値とした．表 5 よりわかるように Mobile PPC では，何も実装していない状態(case1)に比べ移動前(case2)と移動後(case3)ともにスループットの低下はほとんど見られなかった．

Mobile IP は，移動前(case4)ではスループットの低下はほんどないが移動後(case5)では，IP カプセル化によるオーバヘッドと通信経路の冗長により，スループットが 9%ほど低下した．図 5-6 (b)の測定構成では，HA と MN が 1 ホップ分離れている構成であるが，一般的なネットワーク構成では HA を経由することによりラウンドトリップ遅延がさらに増加するとみられるため，スループットがさらに低下することが予想される．

表 5-8 スループットの比較

Table.5-8　The comparison of the throughput

状態		スループット	割合
General	case1	93.237 Mbps	100.000
Mobile PPC	case2	93.236 Mbps	99.999
	case3	93.193 Mbps	99.953
Mobile IP	case4	93.231 Mbps	99.994
	case5	85.202 Mbps	91.382

第5.4節 既存技術のとの比較

表 5-9 に Mobile IP (MIP)，Mobile IPv6 (MIPv6)，LIN6，MAT，Mobile PPC（MPPC）を 6 つの項目で比較した結果を示す. ネットワーク上に設置する第 3 の装置として，Mobile IP における HA，LIN6 における MA，MAT における IMS がそれぞれ必須であり, 新たなネットワーク機器による基盤が必要である. このような装置の存在は，今後普及する P2P 通信の特徴を損なううえ, 一点障害を避けるために二重化等の措置が必要となり，管理負荷が増すという課題が発生する．Mobile PPC では，通信開始時のアドレス管理装置として DDNS を利用するが，DDNS は DNS に機能を拡張したものであり，特殊な第 3 の装置という位置づけではなく，既存環境への適用も容易である.

通信パフォーマンスに影響するものとして通信経路冗長の有無, 一点障害の有無, オーバヘッドがある．Mobile IP はプロキシ方式のため，通信が三角経路になり冗長が発生する. Mobile IPv6 では経路最適化機能が導入されこの問題は解決された. LIN6，MAT，Mobile PPC はエンドツーエンド方式であるため，冗長経路の問題はない．Mobile IPv4 では HA の障害時に全通信が不可能となる一点障害の課題があるが，Mobile IPv6 では，HA の複数設置や散設置[19]が可能となり，この課題は解決された．LIN6，MAT はそれぞれ MN，IMS の二重化が考慮されている.

Mobile IP ではトンネル通信時，Mobile IPv6 は経路最適化通信時にヘッダが追加されることによるヘッダオーバヘッドが発生する．LIN6，MAT，Mobile PPC ではヘッダオーバヘッドは発生せず，一般通信とパケット長は同じである.

Mobile IP は，通信相手ノードに変更を加えない設計となっており，既存端末との通信でも移動透過性を提供できる．Mobile IPv6 でも，Mobile IP と同様に HA を経由した通信を行うことで既存端末に対する移動透過な通信が可能である．LIN6 では，MA をプロキシとして拡張し，既存端末との通信では全ての通信パケットを MA に経由させることにより移動透過性を提供する方式[20]が検討されているが，

LIN6 本来のエンドツーエンド通信の特徴を損なうことになる．MAT，Mobile PPC は既存端末との移動透過性はサポートしていないが，通常の IP アドレスを用いているため通信中に移動しない限り一般通信の開始は可能である．

LIN6 では，独自の IPv6 アドレス形態を用いるのでアドレス体系の取り決めが必要であり，アドレスの利用効率が悪くなるなどの制約がある．

表 5-9 既存技術との比較

Table.5-9 Comparison with existing technologies

	MIP	MIPv6	LIN6	MAT	MPPC
第 3 の装置	HA ×	HA ×	MA ×	IMS ×	なし ○
通信経路	△	○	○	○	○
一点障害	×	○	○	○	○
オーバヘッド	△	△	○	○	○
CN への実装	○	○	○	△	△
アドレス制約	○	○	×	○	○

第6章　Mobile PPC の今後

Mobile PPC には以下のような課題が残されている. エンドツーエンド方式で移動透過性を実現する場合，通信継続の際には悪意あるユーザによるなりすましを防止するため，端末間における確実な認証が必要である．グローバルな環境では，通信相手は不特定多数となるため，事前に認証に必要な共有鍵や証明書を共有することは難しい．Mobile IPv6 では，MN と HA が共有鍵を事前に保持していることを前提とし，CN が HA と MN に宛てて送信した両方のデータを MN が正しく受信することによって共有鍵を生成する仕組みが提案されている．このような方式は，第 3 の装置を使用しない Mobile PPC では適していない．そこで，Mobile PPC による認証機能を実現するために通信開始時に Diffie-Hellman 鍵交換を利用した共有鍵の生成および移動時の成りすましを防止するための認証機構[21]の検討を行っている．また, Mobile PPC はもともと GSCIP[22-23]というアーキテクチャの枠組みの中で移動透過性を実現する手段として考案されたものである．GSCIP においては，あらかじめ同一の通信グループに対して同一の暗号鍵を割り当てる．このような環境下では上記暗号鍵を用いた認証を実現できる．

次に，一般に無線環境でネットワークの移動を行った場合，無線レイヤと L3 が独立してハンドオーバを実行するためパケットロスが避けられない．また，移動ノード同士の通信では, 両者が全く同時に移動した場合に CU が通信相手に到達せず, 移動端末は互いに移動したことを知ることができないという可能性が考えられる．これらの課題はエンドツーエンド方式共通の課題である．これらの解決策として，MN が一時的に新旧 2 つの IP アドレスを保持させることや無線レイヤと Mobile PPC が連携するなどの工夫が今後必要になると考えられる．

第7章　まとめ

本稿では，エンド端末間で移動透過性を実現する Mobile PPC について提案した．Mobile PPC は，エンド端末の IP 層にアドレス変換処理機能を導入する．MN の IP アドレスが変化した時，MN から CN に変化情報を報告し，アドレス変換テーブルを更新する．移動後の通信パケットは上記アドレス変換テーブルに基づき IP 層でアドレス変換する．この方式により，IP アドレスの変化は上位ソフトウェアから隠蔽される．IP アドレスの変換は，IP 層で行われるため，上位ソフトウェアを変更する必要が無く，IP アドレスを単に変換する方式であるためパケット長が変化することがない．特殊な管理サーバが不要であるため，従来研究の方式に比べ導入が容易である．IPv4 において Mobile PPC を FreeBSD 上に実装し，動作の確認と性能測定を行った．その結果，IP アドレス変換による性能の低下がほとんど無いことがわかった．さらに，Mobile IP とスループットの比較を行い，Mobile PPC の処理が通信に与える影響は，Mobile IP に比べて小さいことを示した．移動の際には Mobile PPC 自体のオーバヘッドは少ないものの，アドレス取得によるオーバヘッドを減らす工夫が別途必要であることがわかった．今後は，Mobile PPC に関わる汎用的な認証方式の検討，ロスなし高速ハンドオーバの検討などを検討していくと共に，IPv6 についても本手法の適用を検討する．

謝辞

本論文の執筆および関連研究の遂行にあたり，ご助言とご指導をいただきました名城大学理工学部情報工学科　渡邊晃教授をはじめ，副査をしていただきました名城大学理工学部情報工学科 小川明教授，柳田康幸教授，宇佐見庄五講師，ならびに多くの方々から貴重なコメントをいただきました．ここに厚く御礼申し上げます．

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, vol.J87-DI No.3 P.308-328, March.2004

[2] Perkins. C: “IP Mobility Support for IPv4,” RFC 3344, IETF, August.2002

[3] Perkins. C: “IP Encapsulation within IP,” RFC 2003, IETF, October.1996

[4] Calhoun. P, Perkins. C: “Mobile IP Network AddressIdentifier Extension,” RFC 2794, IETF, March.2000

[5] Perkins. C, Calhoun. P: “Mobile IP Challenge/Response Extensions,” RFC 3012, IETF, November.2000

[6] Montenegro. G: “Reverse Tunneling for Mobile IP,” RFC3024, IETF, January.2001

[7] Johnson. D, Perkins. C, Arkko. J: “Mobility Support in IPv6,” RFC3775, IETF, June.2004.

[8] Pravin Bhagwat, David A. Maltz, and Adrian Segall. MSOCKS+:An architecture for transport layer mobility. ElsevierScience ComputerNetworks, Vol. 39, No. 4, pp. 385–403, July 2002.

[9] Daichi Funato, Kinuko Yasuda, and Hideyuki Tokuda. TCP-R: TCP Mobility Supportfor Continuous Operation. In IEEE International Conference on Network Protocols97, Atlanta, October 1997.

[10] Alex C. snoeren and Hari Balakrishnan: “An End –to-End Approach to Host Mobility,” MIT Laboratory for Computer Science Cambridge MA 02139 , 6th ACM/IEEE International Conference on Mobile Computing and Networking , August.2000

[11] 松岡保静,吉村　健,大矢智之, “エンドツーエンド型 IP ソフトハンドオーバ” 電子情報通信学会論文誌　VOL.J86-B No.8 August 2003.

[12] Ishiyama. M, Kunishi. M, Uehara. K, Esaki. H, Teraoka: “LINA: A New Approach to Mobiity Support in Wide Area Networks,” IEICE Transactions on Communication vol.E84-B No.8 p.2076-2086, August 2001

[13] 相原玲二, 藤田貫大, 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, Dec.2002.

[14] Vixie. P, Thomson. S, Rekhter. Y, Bound. J: “Dynamic Updates in the Domain Name System,” RFC 2136, IETF, April.1997

[15] Droms. R: “Dynamic Host Configuration Protocol,” RFC2131, IETF, March.1997

[16] 小川　猛志, 伊東　匡, ” DHCP をベースとしたシームレスハンドオーバ方法の研究” 電子情報通信学会論文誌　VOL.J88-B No.11 p.2228.

[17] Nick Moore, Greg Daley, ”Fast Address Configuration Strategies for the Mobile Internet, ” ATNAC 2003.

[18] Netperf. http://www.netperf.org

[19] 福田　浩章, ” Mobile IPv6 における分散ホームエージェントの実現”

http://hdl.handle.net/2065/728 2004.
[20] 石山政浩, 國司光宣, 河野通宗, 寺岡文男, “移動体通信プロトコル LIN6 における後方互換性拡張の一方式”, 電子情報通信学会 インターネットアーキテクチャ研究会 論文集, Octover 2002.
[21] 瀬下正樹, 竹内元規, 渡邊晃, ”Mobile PPC における移動端末の認証”, DICOMO2005 シンポジウム論文集, Vol.2005, No.6, pp129-132, Jul.2005.
[22] 鈴木秀和, 渡邊晃, “フレキシブルプライベートネットワークにおける動的処理解決プロトコル DPRP の仕組み”, 情報処理学会研究報告, 2005-CSEC-26, pp. 259-266 (2004)
[23] 鈴木秀和, 渡邊晃, “フレキシブルプライベートネットワークにおける動的処理解決プロトコル DPRP の実装”, 情報処理学会研究報告, 2005-CSEC-28, pp. 199-204 (2005).

研究業績

1). 竹内元規, 渡邊晃, “コネクションを維持した移動端末のP2P通信の提案”, 電気関係学会東海支部連合大会, Oct. 2003.
2). 竹内元規, 渡邊晃, “移動体通信におけるコネクションを維持した通信方式の研究”, 情報処理学会　第 66 回全国大会, Mar.2004.
3). 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, 2004-MBL-030, pp. 17-24 (2004).
4). 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の実装,” 情報処理学会研究報告, 2004-MBL-32,pp.29-35, Mar. 2005.
5). 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 の実装と評価”, マルチメディア, 分散, 協調とモバイルシンポジウム（DICOMO2005）論文集 (査読付き), pp. 125-128 (2005).
6). 竹内元規, 鈴木秀和, 瀬下正樹, 渡邊晃, “移動通信プロトコルMobile PPCの実装とその評価”, 電気関係学会東海支部連合大会, Sep. 2005.
7). 鈴木秀和, 竹内元規, 加藤尚樹, 増田真也, 渡邊晃, “フレキシブルプライベートネットワークを実現するセキュア通信アーキテクチャ GSCIP の提案”, マルチメディア, 分散, 協調とモバイルシンポジウム（DICOMO2005）論文集 (査読付き), pp. 441-444 (2005).
8). 瀬下正樹, 竹内元規, 渡邊晃, “Mobile PPCにおける認証方式の提案”, 電気関係学会東海支部連合大会, Sep. 2004.
9). 瀬下正樹, 竹内元規, 渡邊晃, “Mobile PPC における認証方式の提案”, 情報処理学会　第 67 回全国大会, Mar.2005.
10). 瀬下正樹, 竹内元規, 渡邊晃, “Mobile PPC における移動端末の認証”, マルチメ

ディア，分散，協調とモバイルシンポジウム（DICOMO2005）論文集 (査読付き), pp. 133-136 (2005).
11). 瀬下正樹, 竹内元規, 渡邊晃, “Mobile PPCにおける認証方式の実装に関する検討”, 電気関係学会東海支部連合大会，Sep. 2005.
12). 金本綾子，竹内元規，瀬下正樹，渡邊晃，“IPv6 環境での移動透過性を実現するMobile PPCv6 の検討”，電気関係学会東海支部連合大会，Sep. 2005.
13). 坂本順一，鈴木秀和，竹内元規，渡邊晃，“Mobile PPCを利用した移動ネットワークの提案”，電気関係学会東海支部連合大会，Sep. 2004.
14). 坂本順一，鈴木秀和，竹内元規，渡邊晃，“Mobile PPC を利用したネットワーク単位の移動通信の提案”，情報処理学会　第 67 回全国大会，Mar.2005.
15). 坂本順一，鈴木秀和，竹内元規，渡邊 晃，“Mobile PPC を利用したネットワーク単位の移動透過性の提案”，マルチメディア，分散，協調とモバイルシンポジウム（DICOMO2005）論文集 (査読付き)，pp. 133-136 (2005).
16). 坂本順一，鈴木秀和，竹内元規，渡邊晃，“ネットワーク単位の移動透過性を実現するMobile NPCとその実装”，電気関係学会東海支部連合大会，Sep. 2005.

付録A　Mobile PPC 仕様書

I.　仕様概要

(1) CIT

CIT はハッシュテーブルのチェイン法として設計される．CIT のテーブル構成を図 A-1 に示す．GPIT の配列をレコード（Record），チェインで繋がるレコードをリスト（List）と呼ぶ.

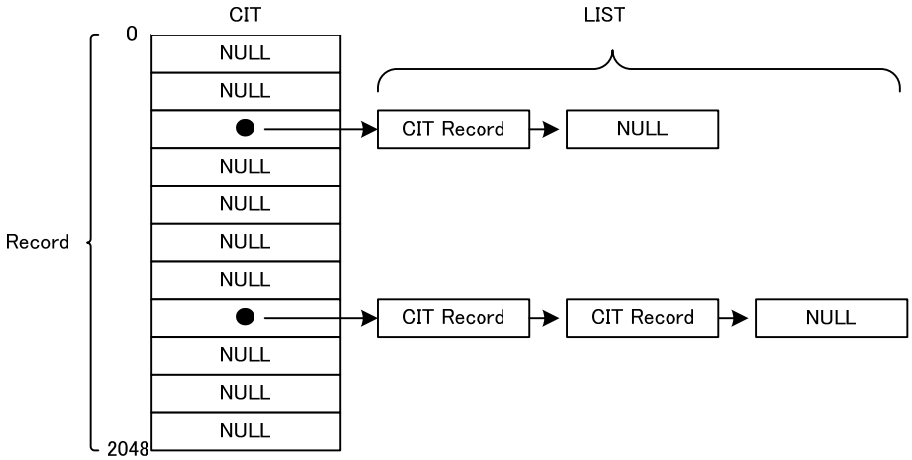

図 A-1　CIT のテーブル構造

(2) CIT Record

GPIT Record は表 A-2 に示す情報から構成されている.

表 A-2　CIT のテーブル構造

フィールド名	内容	フィールド長
sIP	**送信元** IP **アドレス**	4 byte
dIP	**宛先** IP **アドレス**	4 byte
sPrt	**送信元ポート番号**	2 byte
dPrt	**宛先ポート番号**	2 byte
proto	**トランスポート層のプロトコル**	1 byte
trans	**変換処理フラグ**	1 byte
cnt	**削除カウンタ値**	2 byte

tsIP	**変換先送信元** IP **アドレス**	4 byte
tdIP	**変換先宛先** IP **アドレス**	4 byte
tsPrt	**変換先送信元ポート番号**	2 byte
tdPrt	**変換先宛先ポート番号**	2 byte
next	**次のリストへのポインタ**	4 byte

・CIT **サイズ**(MPPC_CIT_SIZE) **:** 2048
・**テーブル長**(MPPC_CIT_HASHPRIME) **:** 2039[CIT **サイズ:** 2048]
・**検索キー** **:** sIP,dIP,sPrt,dPrt,proto **の５つの情報**
・**ハッシュサイズ**(MPPC_HASHLEN) **:** 13byte
・**レコード長** **:** 32byte [sIP～next]

II. 処理概要

(1) 端末起動時の処理

Mobile PPC 対応ノードは，アドレス変換テーブル(CIT)を保持する．端末起動時には，CIT の初期化（CIT サイズ分のメモリ領域を確保）および，以下に示す大域変数の初期化を行う．

◆ CIT **初期化フラグ** mppc_flg_setcit

CIT 初期化フラグ(mppc_flg_setcit)は，CIT の初期化が正常に行われたかどうかを示す．
初期値 ：GFALSE (0)　　CIT初期化後 ：GTURE (1)

表 A-3 CIT 初期化フラグ

変数	データ型	説明
mppc_flg_setcit	**int**	CIT **初期化フラグ**

◆ **移動状態フラグ** mppc_flg_move

移動状態フラグ(mppc_flg_move)は，移動処理の状態を示す．移動処理の状態ごとに，表 A-4 に示すマクロ定数入る．
【移動前の状態】 → 【DHCP パケット検知】 → 【重複アドレスチェッ

ク】　→【移動情報通知状態】　→　【移動前の状態】　と遷移する．

初期値　：MPPC_DETECT_NON (0)

表 A-4　移動状態フラグ

変数	データ型	説明
mppc_flg_move	**int**	**移動状態フラグ**

表 A-5　移動状態を示すマクロ定数

値	マクロ定数	内容
0	**MPPC_DETECT_NON**	**移動前の状態**
1	**MPPC_DETECT_DHCP**	IP **アドレス取得状態**
2	**MPPC_DETECT_ARP**	**重複アドレスチェック状態**
3	**MPPC_DETECT_SEND_CU**	**移動情報通知状態**

◆　**移動前の** IP **アドレス**　mppc_previous_ip

Mobile PPC 対応ノードが移動前に持っていた IP アドレスを示す．移動前のアドレスは以下のよう扱う．

・Mobile PPC 対応ノードは端末起動時に取得したアドレスを大域変数(mppc_previous_ip)に保存
・CIT 更新時には，previous IP から移動前のアドレスを参照
・CIT 更新後，大域変数(mppc_previous_ip)に，新 IP アドレスを保存

初期値　：自端末の起動時のIPアドレス

表 A-6　移動前の IP アドレス

変数	データ型	説明
mppc_previous_ip	**u_long**	**移動前の** IP **アドレス**

◆　**移動後の** IP **アドレス**　mppc_new_ip

Mobile PPC 対応ノードが移動後に取得した IP アドレスを示す．大域変数(mppc_new_ip)を定義し，DHCP などから取得した移動後の IP アドレスを保存する．

初期値　：0.0.0.0

表 A-7　移動後の IP アドレス

変数	データ型	説明
mppc_new_ip	**u_long**	**移動後の** IP **アドレス**

(2) アドレス変換処理

アドレス変換処理は，入出力 TCP/UDP パケット毎に処理を行う．アドレス変換処理フローを図 A-8 に示す．入出力パケットのコネクション識別子をキーとして CIT 検索を行い，その結果により次のような処理を行う．

(1) CITエントリなしの場合
CIT エントリが無い場合は，これから通信が開始されることを示すため，新しく CIT レコードを生成する．このとき生成された CIT レコードの trans は MPPC_TRANS_OFF(0)すなわち，アドレス変換なしの状態となっている

(2) CITエントリあり＆変換なしの場合
この場合は, 通常の通信が行われていることを示す. 処理は行わずに差し戻す.

(3) CITエントリあり＆変換ありの場合
IP アドレス変換処理が必要なパケットであることを示す．CIT レコードの内容にしたがって， パケットの IP アドレスを変換し，それにともなうチェックサムの差分計算を行う．

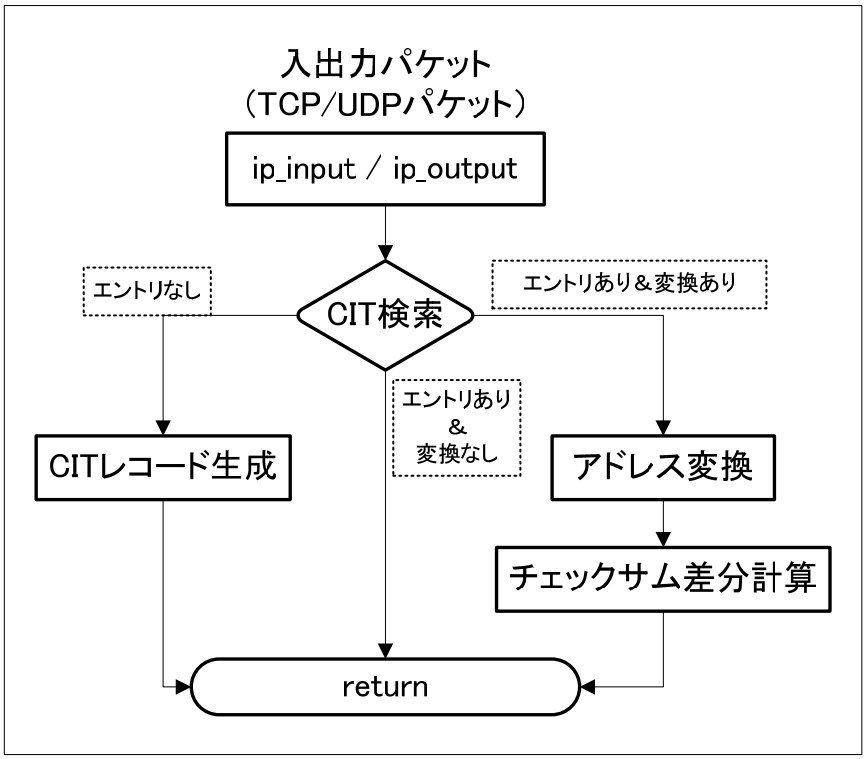

図 A-8　アドレス変換処理フロー

(3) 移動時の処理

図 A-9 に移動時の処理フローを示す．MN 側では CU による移動情報通知処理および CU 応答受信処理，CN 側では CU 受信処理を行う．

(1) 移動情報通知処理

MN がネットワークの移動を行った場合，移動先で DHCP による IP アドレスの取得を行う．MN は，DHCP サーバから IP アドレスを正常に取得したこと示す DHCP ACK パケットから新 IP アドレスを取得し，移動前と移動後のＩＰアドレスを元に移動情報の生成を行う． 移動情報は通信相手毎にまとめられ，CU パケットとして通知する．

(2) CU受信処理

CN は CU パケットを受信すると，CU に含まれる移動情報を取得し，移動前後のコネクション識別子を元に CIT の更新を行う．更新された CIT レコードの trans は，MPPC_TRANS_ON(1)となり，以後の通信では IP アドレス変換が適用

される．CIT 更新後は，CIT の更新が正常に行われたことを通知する CU 応答パケットを生成・送信する．

(2) CU応答受信処理

MN は CU 応答パケットを受信すると，自身の CIT の更新を行う．更新された CIT レコードの trans は，MPPC_TRANS_ON(1)となり，以後の通信では IP アドレス変換が適用される．

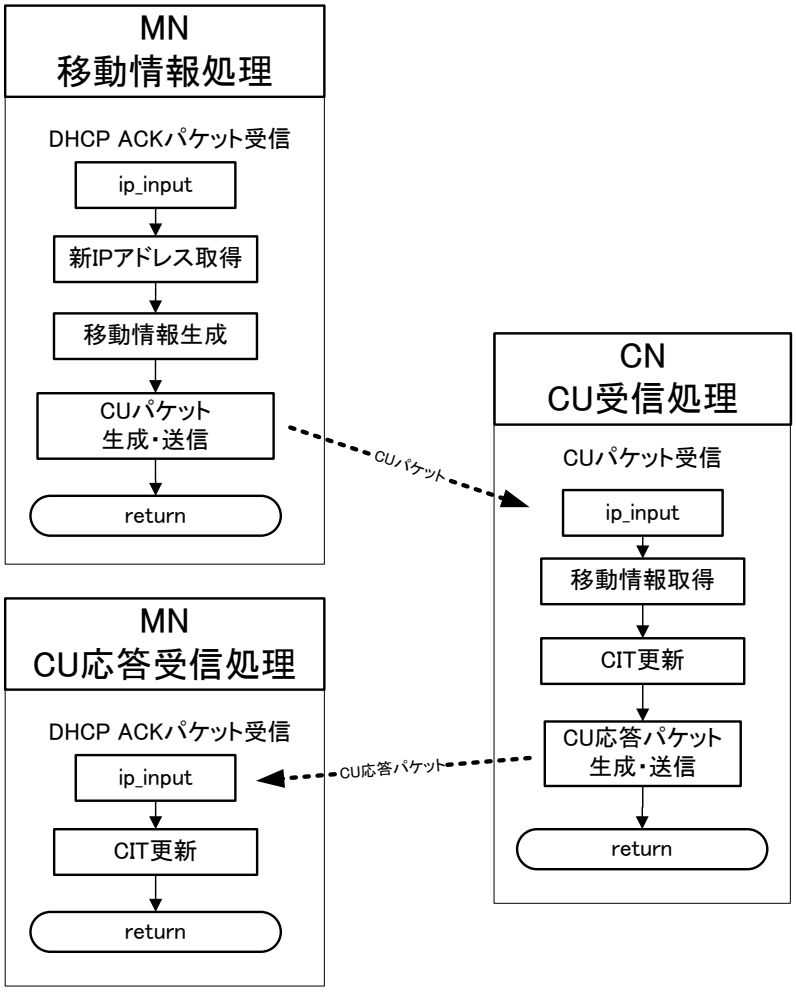

図 A-9　移動時の処理フロー

III. メッセージフォーマット

(1) パケット構成

Mobile PPC の CU, CU 応答は, ICMP Echo Request をベースに定義される, DPRP 制御パケットのオプション部に挿入される． DPRP 制御パケットについては, 本仕様の範囲外でるため別紙「DPRP 仕様書」を参照．メッセージパケットの全体の構造を図 A-10 に示す．Mobile PPC メッセージには，Mobile PPC ヘッダ部と Mobile PPC データ部に分かれている．

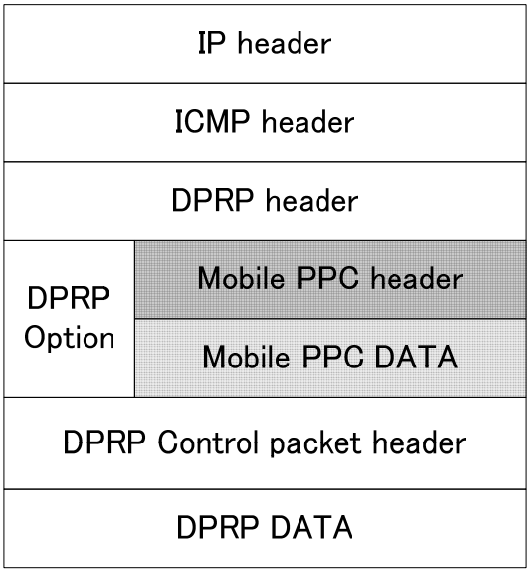

図 A-10　パケット構造

(2) Mobile PPC ヘッダ

Mobile PPC ヘッダフォーマットを図 A-11, ヘッダフィールドを表 A-12 に示す.

ヘッダ長：4 byte (固定)

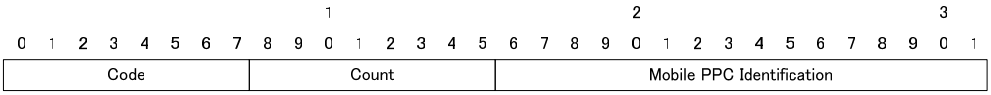

図 A-11　Mobile PPC ヘッダフォーマット

表 A-12　Mobile PPC ヘッダフィールド

フィールド	サイズ	値
type	**1 byte**	**メッセージタイプ**
Count	**1 byte**	**移動情報の数**
Mobile PPC Identification	**2 byte**	**固定** (52428)

(3) CU

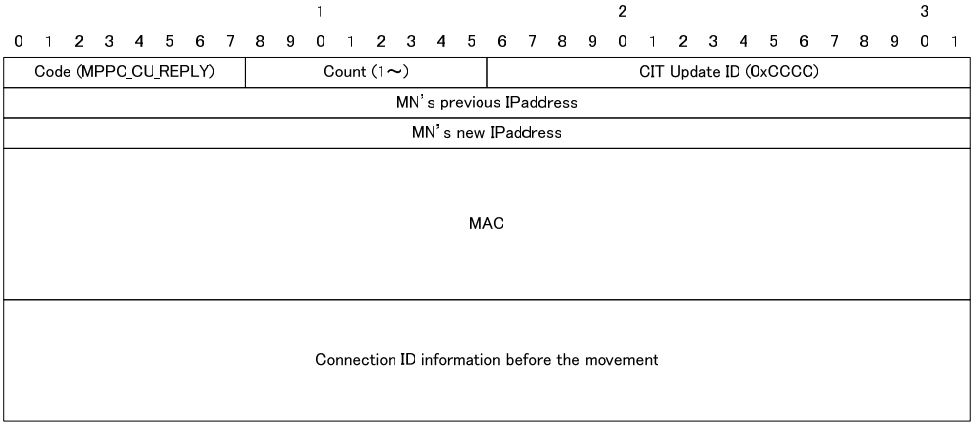

図 A-13　CU メッセージフォーマット

表 A-14　CU メッセージフィールド

フィールド	サイズ	値
type	**1 byte**	MPPC_CU
Count	**1 byte**	**移動情報の数**
Mobile PPC Identification	**2 byte**	**固定** (52428)
MN's previous IPaddress	**4 byte**	MN **の移動前の** IP **アドレス**
MN's new IPaddress	**4 byte**	MN **の移動後の** IP **アドレス**
MAC	**20 byte**	MAC **値**
Connection ID infomation	**可変**	**移動情報**

(4) 移動情報　Connection ID Information

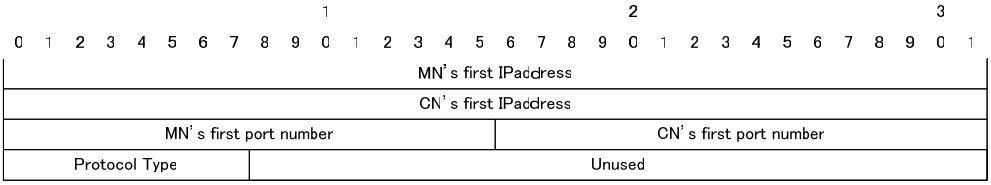

図 A-15　Connection ID Information フォーマット

表 A-16　Connection ID infomation フィールド

フィールド	サイズ	値
MN's first IPaddress	**4 byte**	**通信開始時の** MN **側** IP **アドレス**
CN's first IPaddress	**4 byte**	**通信開始時の** CN **側** IP **アドレス**
MN's first port number	**2 byte**	**通信開始時の** MN **側ポート番号**
CN's first port number	**2 byte**	**通信開始時の** CN **側ポート番号**
Protocol Type	**1 byte**	**プロトコルタイプ**(TCP or UDP)

(5) CU 応答

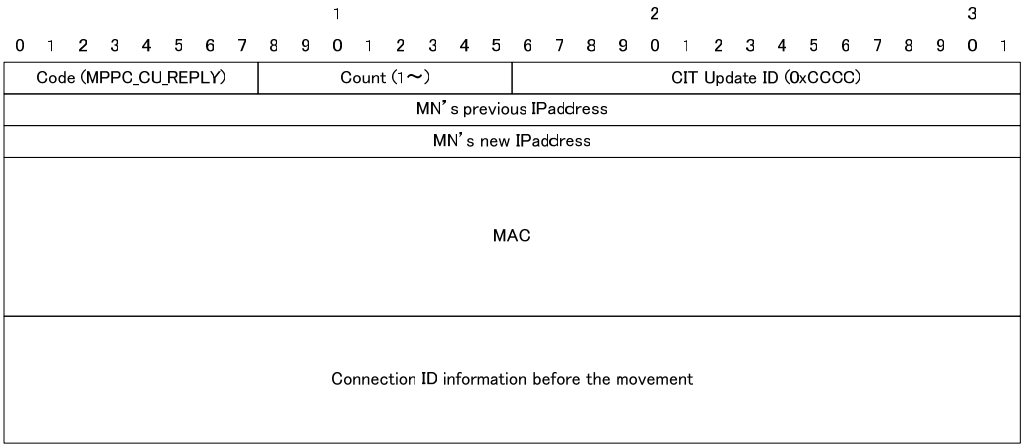

図 A-17　CU 応答メッセージフォーマット

表 A-18　CU 応答メッセージフィールド

フィールド	サイズ	値
type	**1 byte**	MPPC_CU_REPLY
Count	**1 byte**	**移動情報の数**
Mobile PPC Identification	**2 byte**	**固定** (52428)
MN's previous IPaddress	**4 byte**	MN **の移動前の** IP **アドレス**

MN's new IPaddress	**4 byte**	MN **の移動後の** IP **アドレス**
MAC	**20 byte**	MAC **値**
Connection ID infomation	**可変**	**移動情報**

IV. Mobile PPC モジュール構成

(1) ファイル構成

Mobile PPC は表 A-19 に示すファイルから構成されている.

表 A-19 Mobile PPC のファイル構成

ソースファイル	内容
mppc_cit.c	CIT **管理モジュール**
mppc_trans.c	**アドレス変換モジュール**
mppc_detect	**移動管理モジュール**
mppc.h	Mobile PPC **ヘッダ**

(2) CIT 管理モジュール mppc_cit.c

◆ 大域変数

表 A-20 CIT 操作モジュールの大域変数

変数	データ型	説明
hash_cit []	**struct cit ***	CIT **配列**
gcitcnt	**static int**	**カウントの初期値**

◆ 関数

表 A-21 CIT 操作モジュールの関数定義

関数	説明
mppc_cit_check	CIT **レコードを表示するシステムコール** int mppc_cit_check(td, uap)
mppc_cit_add	CIT **レコードを追加するシステムコール**

	int mppc_cit_add(td, uap)
mppc_cit_free	CIT **レコードを削除するシステムコール** int mppc_cit_free(td, uap)
mppc_hash_cid	**入力** CID **からハッシュ値を計算する** u_short mppc_hash_cid(CID *cid)
mppc_check_hash_cit	**入力** CID **から該当** CIT **レコードが存在するかどうかをチェックする** int mppc_check_hash_cit(CID *cid, int no)
mppc_reg_cit	CIT **レコードを生成する** int mppc_reg_cit(CID *cid, CID *t_cid, u_char trans)
mppc_cit_serch	**入力** CID **を元に** CIT **レコードを検索** int mppc_cit_serch(CID *cid)
mppc_cit_serch_by_hash	**入力ハッシュ値を元に** CIT **レコードを検索** int mppc_cit_serch_by_hash(u_short hashval, int no, CID *cid, CID *t_cid)
mppc_cit_serch_by_pkt	**入力パケットを元に** CIT **レコードを検索** int mppc_cit_serch_by_pkt(struct ip *ip, char *nxthdr)
mppc_cit_make_moveinfo	**移動情報を生成** void mppc_cit_make_moveinfo(u_long new_ip)
mppc_rcv_cit_update	CU **受信時に** CIT **レコードを更新** void mppc_rcv_cit_update(CID *old_cid, *new_cid)
mppc_print_cit	**該当する** CIT **レコードを表示する** void mppc_print_cit(u_short citent, struct cit *p, int list_no)
mppc_print_address	IP **アドレス表示** void mppc_print_address(char name[256] , u_long *ip)
mppc_set_table	CIT **を初期化する** int mppc_set_table(void)

表 A-22　CIT 操作モジュールの関数引数

関数	引数			
	変数	データ型	説明	W/R
mppc_cit_check	td	struct td *	システムコール番号	R
	uap	struct uap *	引数(strcut cid)	R
mppc_cit_add	td	struct td *	システムコール番号	R
	uap	struct uap *	引数(strcut cid)	R

mppc_cit_free	td	struct td *	システムコール番号	R
	uap	struct uap *	none	R
mppc_hash_cid	cid	CID *	コネクション識別子	R
mppc_check_hash_cit	cid	CID *	コネクション識別子	R
	no	int	ハッシュリスト番号	R
mppc_reg_cit	cid	CID *	コネクション識別子	R
	t_cid	CID *	変換先コネクション識別子	R
	trans	u_char	変換処理フラグ	R
mppc_cit_serch	cid	CID *	コネクション識別子	R
mppc_cit_serch_by_hash	hashval	u_short	ハッシュ値	R
	cid	CID *	コネクション識別子	R
	t_cid	CID	変換先コネクション識別子	R
mppc_cit_serch_by_pkt	ip	struct ip *	IP ヘッダ	R
	nxthdr	char *	上位ヘッダ	R
mppc_cit_make_moveinfo	new_ip	u_long	DHCP で取得した IP アドレス	R
mppc_rcv_cit_update	old_cid	CID *	コネクション識別子	R
	new_cid	CID *	コネクション識別子	R
mppc_print_cit	name[256]	char	コメント文	R
	ip	u_long	IP アドレス	R

(3) アドレス変換モジュール mppc_trans.c

◆ 関数

表 A-23　アドレス変換モジュールの関数定義

関数	説明
mppc_trance	**主処理** void mppc_trance(struct ip *ip, char *nxthdr, int mode)
mppc_nat	IP **アドレス付け替え処理** void mppc_nat(struct ip *ip, char *nxthdr, CID *cid, int mode)
mppc_cksum	**チェックサム差分計算** void mppc_cksum(u_char *cksum, u_char *optr, int olen, u_char *nptr, int nlen)

表 A-24　アドレス変換モジュールの関数引数

関数	引数			
	変数	データ型	説明	W/R
mppc_trance	ip	struct ip *	IP ヘッダ	W
	nexthdr	char *	上位ヘッダ	W
	mode	int	入力/出力	
mppc_nat	ip	struct ip *	IP ヘッダ	W
	nexthdr	char *	上位ヘッダ	W
	cid	CID *	コネクション識別子	R
	mode	int	入力/出力	R
mppc_cksum	cksum	u_char *	TCPorUDP チェックサム	W
	optr	u_char *	入力データ	R
	olen	int	入力データ長	R
	nptr	u_char *	出力データ	W
	nlen	int	出力データ長	W

(4) 移動管理モジュール mppc_detect.c

◆ 関数

表 A-25　移動管理モジュールの関数定義

関数	説明
mppc_message_type	Mobile PPC ヘッダのタイプ値チェック int mppc_message_type(mbuf *m0)
mppc_isdprp_opt	DPRP コントロールヘッダのオプションタイプ値チェック int mppc_isdprp_opt(struct mbuf *m0)
mppc_detect_dhcp	DHCP パケット解析 int mppc_detect_dhcp(struct *ip, char *nxthdr, mbuf *m)
mppc_call_dprp	DPRP 呼び出し int mppc_call_dprp(MOVE_INFO *minfo)
mppc_make_cu	CU 生成処理 int mppc_make_cu(struct mbuf *m0, struct mbuf *m, int *len)
mppc_rcv_cu	CU 受信処理

	int mppc_rcv_cu(struct mbuf *m0)
mppc_make_cu_reply	CU **応答生成処理** int mppc_make_cu_reply(struct mbuf *m0, struct mbuf *m, int *len)
mppc_rcv_cu_reply	CU **応答受信処理** int mppc_rcv_cu_reply(struct mbuf *m0)
mppc_route_add	**ルーティングテーブル修正処理** void mppc_route_add(u_long new_mn_addr, u_long old_mn_addr)

表 A-26　アドレス変換モジュールの関数引数

関数	引数			
	変数	データ型	説明	W/R
mppc_message_type	m0	struct mbuf *	**受信パケット**	R
mppc_isdprp_opt	m0	struct mbuf *	**受信パケット**	R
mppc_detect_dhcp	ip	struct ip *	IP **ヘッダ**	R
	nexthdr	char *	**上位ヘッダ**	R
	m	struct mbuf *	**受信パケット**	R
mppc_call_dprp	minfo	MOVE_INFO *	**移動情報**	R
mppc_make_cu	m0	struct mbuf *	**受信パケット**	R
	m	struct mbuf *	**送信パケット**(CU **パケット**)	R
	len	int *	CU **パケット長**	W
mppc_rcv_cu	m0	struct mbuf *	**受信パケット**(CU **パケット**)	R
mppc_make_cu_reply	m0	struct mbuf *	**受信パケット**	R
	m	struct mbuf *	**送信パケット**(CU **応答パケット**)	R
	len	int *	CU **応答パケット長**	W
mppc_rcv_cu_reply	m0	struct mbuf *	**受信パケット**(CU **応答パケット**)	R
mppc_route_add	new_mn_addr	u_long	MN **の移動後の** IP **アドレス**	R
	old_mn_addr	u_long	MN **の移動前の** IP **アドレス**	R

(5) Mobile PPC ヘッダ

◆ マクロ定数

表 A-27　Mobile PPC ヘッダタイプ

値	マクロ定数	内容
1	MPPC_DNS	ドメインネーム検証
2	MPPC_COOKIE	クッキー
3	MPPC_COOKIE_REPLY	クッキー応答
4	MPPC_DH	DH 交換
5	MPPC_DH_REPLY	DH 交換応答
6	MPPC_CU	CU
7	MPPC_CU_REPLY	CU 応答

表 A-28　送受信フラグ

値	マクロ定数	内容
1	MPPC_RCV	受信パケット
2	MPPC_SND	送信パケット

表 A-29　trans 値

値	マクロ定数	内容
0	MPPC_TRANS_ON	変換する
1	MPPC_TRANS_OFF	変換しない

表 A-30　CIT 検索結果

値	マクロ定数	内容
0	MPPC_CIT_NON	該当レコードなし
1	MPPC_CIT_HIT	該当レコードあり
2	MPPC_CIT_HIT_TRANSOFF	該当レコードあり＆変換先なし
3	MPPC_CIT_HIT_TRANSON	該当レコードあり＆変換先あり

表 A-31　移動状態フラグ値

値	マクロ定数	内容
0	MPPC_DETECT_NON	移動前
1	MPPC_DETECT_DHCP	DHCP パケット受信状態
2	MPPC_DETECT_ARP	重複アドレス検知 ARP 受信状態
3	MPPC_DETECT_SEDN_CU	移動情報通知状態

表 A-32　CIT に関するマクロ定数

値	マクロ定数	内容
2048	**MPPC_CIT_SIZE**	CIT **テーブルサイズ**
13	**MPPC_CIT_HASLEN**	**ハッシュ長**
1021or37	**MPPC_CIT_HASHPRIME**	**ハッシュ素数**

表 A-33　長さ定義

値	マクロ定数	内容
sizeof (struct mppc_hdr)	**MPPC_HLEN**	Mobile PPC **ヘッダ長** (4 byte)
sizeof (struct node_data)	**MPPC_NODEDATALEN**	NODE **データ長** (28 byte)
20	**MPPC_MACLEN**	MAC **長** (20 byte)

◆ 構造体

表 A-34　構造体定義

構造体	型定義	説明
cit	**CIT**	CIT **レコード**
mppc_hdr	**MPPC_HDR**	MPPC **ヘッダ**
node_data	**NODE_DATA**	**ノードデータヘッダ**
move_info	**MOVE_INFO**	**移動情報**

表 A-35　CIT 構造体

メンバ	データ型	説明
saddr	**u_long**	**送信元** IP **アドレス**
daddr	**u_long**	**宛先** IP **アドレス**
sport	**u_short**	**送信元ポート番号**
dport	**u_short**	**宛先ポート番号**
proto	**u_char**	**プロトコル番号**
trans	**u_char**	**変換処理フラグ**
cnt	**u_char**	**カウンタ値**
tsaddr	**u_long**	**変換先送信元** IP **アドレス**
tdaddr	**u_long**	**変換先宛先** IP **アドレス**
tsport	**u_short**	**変換先送信元ポート番号**

tdport	**u_short**	**変換先宛先ポート番号**
next	**struct cit ***	**次のリストへのポインタ**

表 A-36　mppc_hdr 構造体

メンバ	データ型	説明
type	**u_char**	Mobile PPC **メッセージタイプ**
cu_cnt	**u_char**	**移動情報の数**
cu_id	**u_short**	Mobile PPC Identification

表 A-37　 node_data 構造体

メンバ	データ型	説明
mn_pre_addr	**u_long**	MN **の移動前の** IP **アドレス**
mn_new_addr	**u_long**	MN **の移動後の** IP **アドレス**
mppc_mac [MPPC_MACLEN]	**u_char**	MAC **値**

表 A-38　move_info 構造体

メンバ	データ型	説明
cnt	**int**	**移動情報の数**
daddr	**u_long**	CN **の** IP **アドレス**
h_list [15]	**int**	**ハッシュ値**

(6) ファイル構成図

表 A-39　モジュール構成図の説明

		
白抜き	Mobile PPC モジュールで定義 サブモジュールを含む	Mobile PPC モジュールで定義 サブモジュールを含まない
色付き	Mobile PPC 以外のモジュールで定義．サブモジュールを含む	Mobile PPC 以外のモジュールで定義．サブモジュールを含まない

◆ CIT 管理モジュール

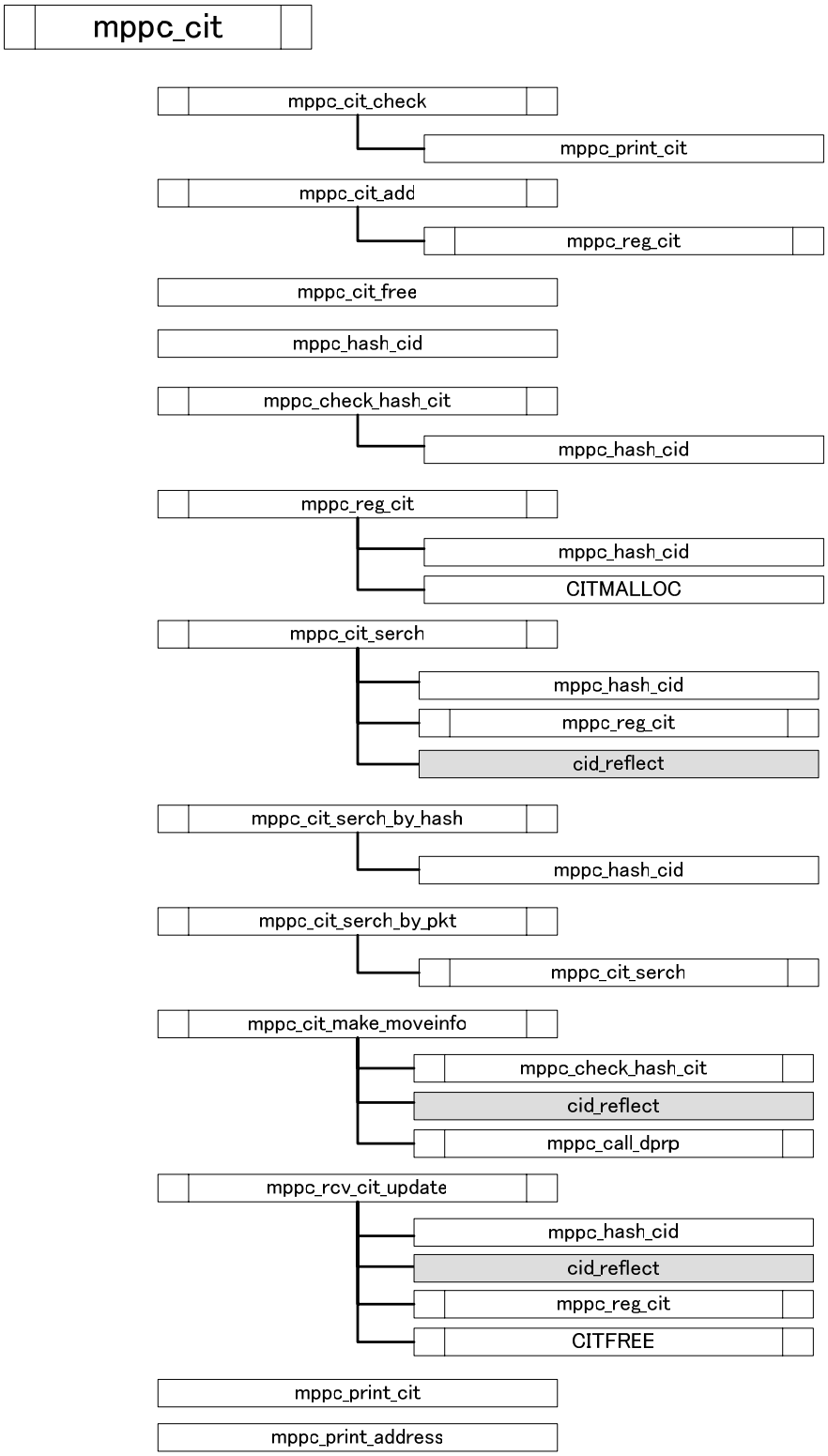

図 A-40　CIT 管理モジュール構成

◆ アドレス変換モジュール

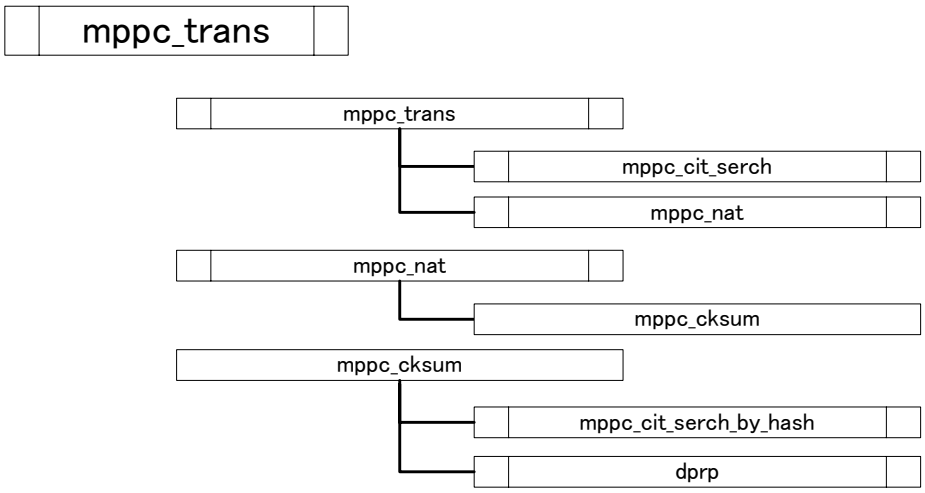

図 A-41　アドレス変換モジュール構成

◆ 移動管理モジュール

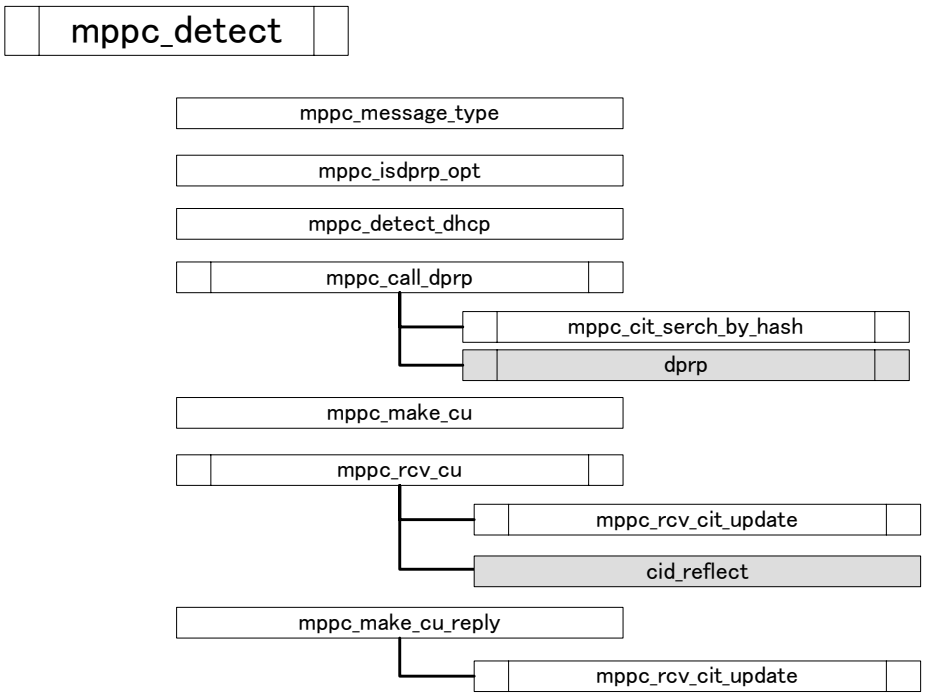

図 A-42　移動管理モジュール構成