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

竹内　元規　　鈴木　秀和　　渡邊　晃
名城大学大学院理工学研究科

Implementation and its evaluation on Mobile PPC realizing end-to-end mobility

Motoki Takeuchi　　Hidekazu Suzuki　　Akira Watanabe
Graduate School of Science and Technology, Meijo University

1.　はじめに

無線 LAN の普及により，ノート PC や PDA を初めとした多くのモバイル端末がネットワークに接続され，いつでもどこでも通信が可能な環境が整備されつつある．この様な環境では，通信中に移動を行っても，通信に影響を与えない移動透過性が要求される．TCP/IP では，ノードを識別する IP アドレス自体に位置の情報を含んでいるため，移動ノードがネットワークを移動すると異なる IP アドレスが必要となる．上位層では IP アドレスが異なると違う通信と見なすため，通信を継続することができなくなる[1].

これまでに提案されている移動透過性の研究には，Mobile IP[2]，Mobile IPv6[3]，LIN6(Location Independent Networking for IPv6)[4]などがある．Mobile IP は完成された技術であるが，通信経路の冗長やヘッダの追加によるオーバヘッド，HA という特殊な装置が必要となり，HA が複数設置できないことによる一点障害などの問題点が指摘されている．Mobile IPv6 では，経路最適化機能が標準で追加されたことで，経路の冗長，オーバヘッドなどの課題は緩和しているが，通信開始時には HA を経由するルーティングを行うため，HA が必須となることに変わりない．

LIN6 では，ノード識別子と位置指示子を分離させることで，IP アドレス変化時の問題を解決する．しかし，IPv6 のアドレス構造を利用した縮退アドレスモデルを適用しているため，アドレスの利用効率が低下する．独自のアドレス体系を持つため，ノード識別子のグローバルユニークな割当てが必要となるという課題がある．また，ノード識別子と IP アドレスの対応を保持する位置管理サーバ(MA)が必要となる．

移動透過な通信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(初期 IP アドレスの解決)と，通信中に IP アドレスが変わった場合に通信を継続できる方法(継続 IP アドレスの解決)の２つを解決する必要がある．著者らは，初期 IP アドレスの解決と継続 IP アドレスの解決の２つを明確に分離してこの問題の解決をはかっている．

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)[5]という技術が既に実用となっており，これを利用する．MN が移動先で新しく取得した IP アドレスを DNS に登録しておくことで，CN は MN のホスト名により IP アドレスを知ることができ，通信開始が可能になる．

継続 IP アドレスを解決する手段として，我々が提案している Mobile PPC(Mobile Peer to Peer Communication)[7]を適用する．モバイル端末が通信中にネットワークを移動し IP アドレスが変化した場合には，両端末においてアドレス変換処理を行うことによって上位ソフトウェアに影響を与えないまま通信を継続させることができる．また，エンドエンドによる実装のみで，途中のルータ等に一切変更を与えずに導入ができる利点がある．

Mobile PPC の実装は，FreeBSD のカーネル部に既存の処理へ影響を与えないようにモジュールを組み込むことで行っており，実験環境で基本動作の検証を行い，移動透過な通信ができることを確認済みである．また，性能測定を行った結果，Mobile PPC の処理によるオーバヘッドはほとんど発生しないことが分かった．本稿では，Mobile PPC の実装方式とその性能評価結果について述べる．

2.　Mobile PPC

2.1.　動作概要

Mobile PPC では，通信を行っているエンド端末の IP 層に移動の通知処理，アドレス変換処理を挿入し，これをコネクションごとに処理することで継続 IP アドレスを解決する．以下に，初期 IP アドレスは DDNS で解決されることを前提として Mobile PPC の動作概要を述べる．

TCP/UDP におけるコネクション識別子には，通信を行っている両端末の IP アドレスとポート番号の組，プロトコル

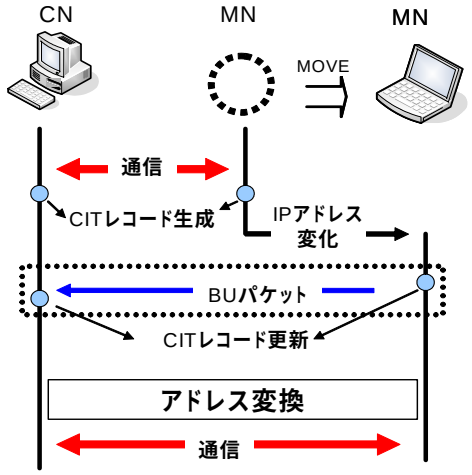


図 1　移動情報の通知
Fig.1 The notice of move information

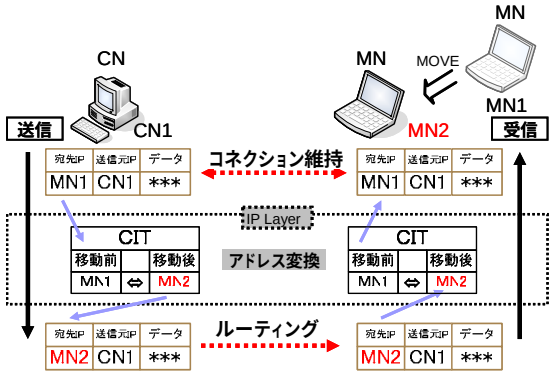


図 2　アドレス変換の例
Fig.2 The example of address translation

番号の５つの情報が使われている．そこで，通信開始時にコネクション識別子を保存しておき，移動により IP アドレスが変化した場合には, IP 層でパケットごとに通信開始時のコネクション識別子に一致するように通信パケットの IP アドレス変換することで，上位層にたいして同一の通信だと見せかけることができる．

図１に Mobile PPC による移動情報の通知方法，図 2 に MN の IP アドレスが MN1 から MN2 へと変化した場合のアドレス変換の例を示す．エンド端末はそれぞれアドレス変換のために移動前後のコネクション識別子の対応関係を記すテーブル（Connection ID Table；以下 CIT）を管理する必要がある．移動ノード(MN)と通信相手ノード(CN)で通信が開始されると，両端末で CIT レコードが生成される．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(Dynamic Host Configuration Protocol)[6]などによって新しく IP アドレスを取得する．ここで MN は，自身の保持する CIT を更新するとともに，移動情報を Binding UPDATE(以下, BU)パケットとして生成し, CN に通知する．CN は，BU により通知された情報を元に自身の CIT を更新する．移動情報通知後は，更新された CIT レコードに従い送受信パケットに対し，IP 層にて IP アドレスの書き換え処理を行う．CN から送信されたパケットの宛先は，CIT を参照し MN の移動前の IP アドレス MN1 から移動後の IP アドレス MN2 へ変換される．このパケットを受信した MN は，自身の CIT を参照し，パケットの宛先を移動後の IP アドレス MN2 から移動前の IP アドレス MN1 へ変換を行い上位層へ渡す．MN から送信される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，上位層にはその変化を隠蔽するため移動前後においてコネクションを維持させることが可能となる．

2.2. アドレス変換の適用範囲

図 3 にアドレス変換の適用範囲を示す．アドレス変換処理は，移動前後で通信が行われていたセッションに対して行われる．実際の通信が始まってしまうと上位層では同じ IP アドレスを使い続けるため，移動を繰り返した場合でも，通信開始時の IP アドレスと移動先で取得した IP アドレスによる 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，移動後に取得した 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，セッション毎に上位層で認識する IP アドレスが異なる状態となる．

2.3. Mobile PPC を利用することによる利点

Mobile PPC は，エンド端末の通信をコネクション単位で識別するので TCP/UDP に関わらず通信の継続ができる．また，エンドエンドによる実装のみでコネクション維持ができるため，特殊な装置は不要で，ルータ等に一切変更を与える必要がない．

IP アドレスの変換は, 既存の処理にほとんど変更を加えずに実装ができ，Mobile PPC を実装したことによるオーバヘッドは極めて少ない．また IP 層で全ての処理を終えるため TCP より上位の層には影響をあたえない．

移動後に継続する通信は，通信開始時と移動後の IP アドレスによる変換のみとなるので，パケット長が変化することがなく通信経路の冗長は生じない．通常の IP アドレスを使用するので，原理的に IPv4 でも IPv6 でも適用可能である．

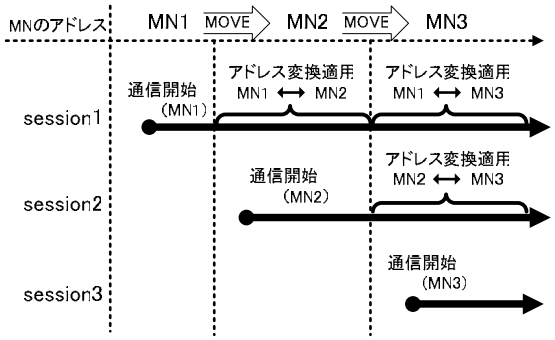


図 3 アドレス変換の適用範囲

Fig.3 Coverage of address translation.

2.4. Mobile PPC 未実装端末との通信

Mobile PPC で用いる IP アドレス及びプロトコルは，一般通信で用いられるものと全く同じである．このため，Mobile PPC を実装していない一般端末とも通常通りに通信ができる．ただし，Mobile PPC 対応ノードが一般端末と通信中に移動した場合は，移動透過な通信のサポートはできないため通信は途絶える. この場合, 一般端末は DDNS により Mobile PPC 対応ノードの IP アドレスを知ることができるので，移動後に一般端末側または移動端末から通信を開始することは可能である．

2.5. ルーティングテーブルの更新

FreeBSD では IP パケット送信時のルーティング情報検索高速化のため、コネクション確立時に作成されるソケットにはルーティング情報がキャッシュされている. このため, MN と CN が同一サブネット上で通信をしている時に，MN が別のサブネットに移動した場合，CN のルーティンテーブル内には，MN が同一サブネット上に存在しているという情報が残ってしまい正しくパケットの送信を行うことができない．そこで，Mobile PPC では CN が BU 受信処理を完了後に自身のルーティングテーブルに対し，BU を送信した MN へのルーティングをデフォルトゲートウェイ経由とするよう更新する．MN 側では，DHCP による IP アドレス取得処理によりルーティングテーブルが更新されるのでこの処理を行う必要はない．

3. 実装

3.1. モジュール構成

実装対象となる OS はオープンソースで, IP 層に関する情報や処理内容の資料が多い FreeBSD を採用した．

Mobile PPC の機能を実現するためのモジュール構成を図 3 に示す．IP 層に組み込まれるものとしてアドレス変換モジュール，移動管理モジュール，CIT 操作モジュール，アプリケーションレベルで動作するものとして CIT 削除デーモンがある．既存の処理に変更を加えないようパケット受信時には IP 入力関数である ip_input, パケット送信時には IP 出力関数である ip_outpu 内で Mobile PPC を呼び出し，処理を終えたら差し戻す形をとっている．

3.2. CIT 操作モジュール

CIT 管理モジュールは，アドレス変換・移動通知モジュールの双方から呼ばれ，CIT レコードの検索・生成・更新を実行する．CIT は，通信開始時および移動時のコネクション識別子（sIP/dIP, sport/dport, proto, tsIP/tdIP, tsport/tdport），変換処理フラグ(trans)，カウンタ値(cnt)の 11 個の情報を保持する．CIT の構造を図 5 に示す．CIT はハッシュテーブル

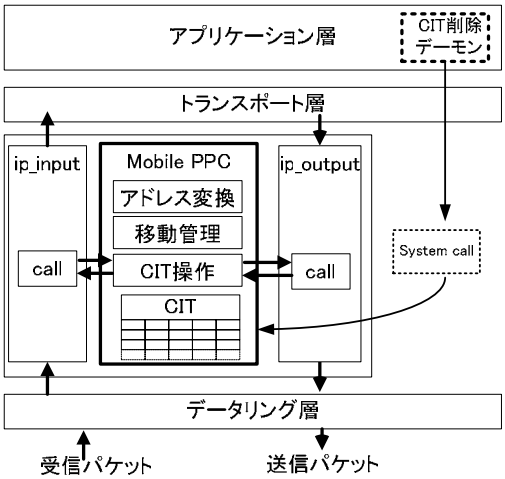


図 4　モジュール構成
Fig.4 Processing in IP layer

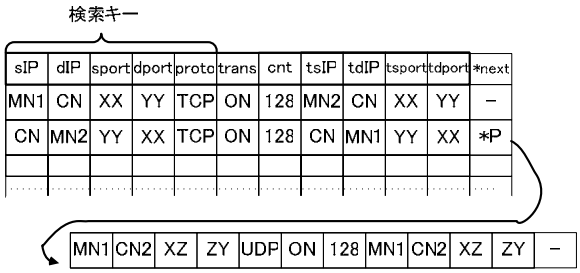


図 5　CIT の構造
Fig.5 CIT Construction

として実装し，検索キーは通信開始時のコネクション識別子となる sIP,dIP,sport,dport,proto の５つの情報のハッシュ値である．また，ハッシュ検索アルゴリズムにはチェイン法を用いおり，レコードの最後尾には衝突回避用に次の CIT レコードを示すフィールド追加されている．CIT レコード長は 32byte，CIT のサイズは 2048 レコードである．端末起動時に CIT サイズ分のメモリ領域を確保する．

3.3.　CIT 削除デーモン

CIT レコード内のカウンタ値は，CIT 削除デーモンにより定期に的にデクリメントする．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し，そのレコードを削除する．レコードが削除される前に通信が発生したらカウンタ値を初期化する．

3.4.　アドレス変換モジュール

アドレス変換モジュールは，TCP/UDP パケットの送信および受信パケット毎に呼び出されるモジュールである．アドレス変換モジュールの処理フローを図 6 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 OFF 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．

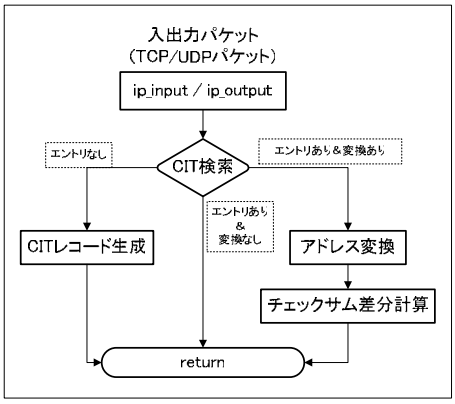


図 6　アドレス変換モジュールの処理フロー
Fig.6 Processing flow of address translation module

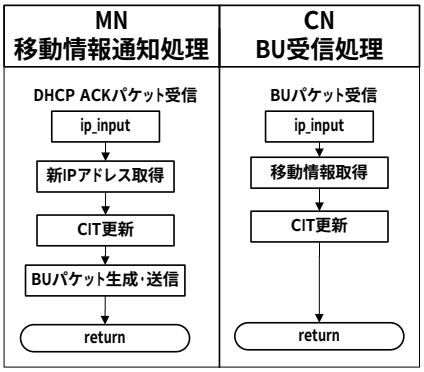


図 7　移動管理モジュールの処理フロー
Fig.7 Processing flow of movement control module

3.5.　移動管理モジュール

図 7 に移動管理モジュールの処理フローを示す． MN 側では BU による移動情報の通知処理，CN 側では BU 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．移動管理モジュールは，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 CIT の更新を行う．このとき更新された CIT レコードの trans は ON すなわち変換ありの状態となり，以後の通信では IP アドレス変換が適用される．更新された CIT レコードの情報は通信相手毎にまとめられ，BU パケットの移動情報として付加される．

BU パケットの送信元 IP アドレスは，移動後の IP アドレスが設定される必要があるが，DHCP ACK パケットを受信した時点では， 新しく取得した IP アドレスの内部設定が完了していない．そのため，移動情報通知処理は，IP アドレス取得処理の最後に行われる重複アドレスチェックの ARP パケットをトリガーとして実行する．

(2) BU 受信処理

CN は BU パケットを受信すると，BU 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 レコードは，MN における処理と同様に trans が ON となり，以後の通信では IP アドレス変換が適用される．

4. 性能評価

4.1. 移動透過性の確認

移動透過性の確認を図 8 に示す実験環境で行った．MN と CN の OS は FreeBSD で Mobile PPC を実装している．MN の CPU は Celeron 2GHz，メモリが 256M で IEEE802.11b で実験ネットワークに接続している．CN の CPU は Pentium 2.4GHz，メモリが 256M，NIC は 100BASE-T である．

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 のネットワークを移動させた．DHCP により新しく IP アドレスを取得して，IP アドレスが変化したがその後もデータ通信が継続していることを確認した．

4.2. １パケットごとの処理時間

Mobile PPC 実装時で移動前（アドレス変換なし），移動後（アドレス変換あり）の１パケットごとの内部処理時間を表 1 に示す．内部処理時間の測定には Pentium Time Stamp Counter を用いて 100 パケットごとの処理時間の平均を測定した．表 1 より，Mobile PPC のアドレス変換の適用にかかる時間は 0.5μ秒ほどであることがわかった．

4.3. FTP による性能評価

Mobile PPC を実装した場合，アプリケーションに与える影響がどの程度あるかを確認するため，FTP による転送時間の違いを測定した．

Mobile PPC を実装していない状態，Mobile PPC を実装しアドレス変換をしていない状態，アドレス変換をしている状態のそれぞれの場合で，ＭＮがＣＮから 50M のファイルを FTP でダウンロードしたときの実行時間を比較したものを表 2 に示す．Mobile PPC を実装していない状態を基準とした場合，Mobile PPC 実装時でアドレス変換なしの状態では同じ時間となり，アドレス変換をしている状態でも 0.1%程度の処理時間の増加であることがわかった．このことにより，Mobile PPC によるオーバヘッドは，極めて小さくアプリケーションにはほとんど影響がないと考えられる．

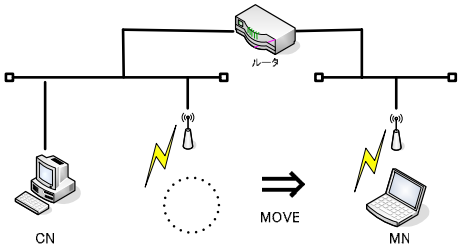


図 8　実験環境

Fig.8 Experimental environment

表 1　１パケットごとの処理時間

table.1 The processing time of every 1 packet

アドレス変換の有無	変換なし	変換あり
平均時間　[μ秒]	1.04	1.57

表 2　FTP によるダウンロード時間

table.2 Download time by FTP

状態	ダウンロード時間	割合
Mobile PPC 未実装	87.31［秒]	1
アドレス変換なし	87.31［秒]	1
アドレス変換あり	87.40［秒]	1.001

5. Mobile PPC の課題

Mobile PPC の課題として，移動時の認証，移動時のパケットロス，移動ノード同士の通信における同時移動などが挙げられる．移動時にはセキュリティの観点から端末間の確実な認証が必要である．現時点では，エンド端末間であらかじめ共有鍵を保持している環境を想定し，この共通鍵を利用した認証を行っているが，グローバルな環境での認証機能が定義されていない．そのため，移動時の成りすましを防止するための認証機構を定義していく必要がある．

端末がネットワークを移動し通信が再開されるまでの間にはパケットロスの発生が考えられるため，より高速なハンドオーバ処理が必要となる．IPv4 環境では，MN が単独でネットワークを移動したことを検知する機能がなく，DHCP による IP アドレスの取得のトリガーの規定が困難である．これらの解決には，無線レイヤとの連携により解決をはかっていく必要がある．

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 BU メッセージが伝わらず通信が継続できなくなる可能性が考えられるため，一時的に新旧 IP アドレスの両方を保持するような対策が必要である．

6. おわりに

本稿では，端末の移動後に IP 層で IP アドレス変換を行い IP アドレスの変化を上位ソフトウェアから隠蔽し，モバイル端末の移動透過性を実現する通信方式 Mobile PPC について述べた．また，Mobile PPC を IPv4 上で実装を行い，移動透過な通信ができること，オーバヘッドが極めて少ないことを確認した．

今後は，課題の解決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．

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-D-I, No.3，pp.308-328(2004)

[2]. Perkins,C. ： IP Mobility Support for IPv4, RFC3344,IETF,Aug.2002

[3]. Johson,D.B. and Perkins,C. : IP Mobility Support in IPv6 , Internet-draft , TETF , Nov.2002

[4]. Ishiyama , M. , Kunishi,M., Uehara,K, Esaki.H,and Teraoka .F , ：LINA : A New Approach to Mobiity Support in Wide Area Networks , IEICE Trans. Commun. , Vol.E84-B , No.8 , PP.2076-2086 (2001)

[5]. Vixie (Ed.), P., Thomson, S., Rekhter, Y. and J. Bound, ： "Dynamic Updates in the Domain Name System", RFC 2136,April 1997.

[6]. R. Droms, “ Dynamic Host Configuration Protocol ”, RFC2131，March 1997.

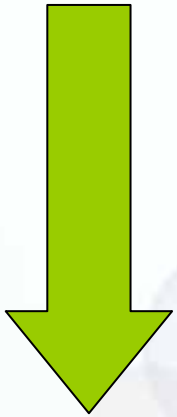
[7]. 竹内元規，渡邊晃，“ モバイル端末の移動透過性を実現する Mobile PPC の実装, ” 情報処理学会研究報告，2004-MBL-32, pp.29-35, Mar. 2005.

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

名城大学大学院理工学研究科
竹内　元規
鈴木　秀和
瀬下　正樹
渡邊　晃

研究背景

- ユビキタスネットワークの構築
  - 無線ネットワークを主体として，いつでもどこでも通信可能な環境が整備されつつある



- 移動を行うとIPアドレスが変化
  - 移動ノードのIPアドレス？
  - 上位層で，別の通信と見なされてしまう

通信中に移動を行っても通信に影響を与えない

移動透過性の研究 Mobile IP , LIN6，･･･，

移動通信プロトコルへの要求

- 今後のユビキタス社会では

P2P通信の要求が増加　　**個人間の通信が主体**

  - P2P通信の特徴を損なわない方式
  - 既存の仕組みをできるだけ変えない方式

特殊な第3の機器を使用することなく，P2Pで移動透過性を実現する

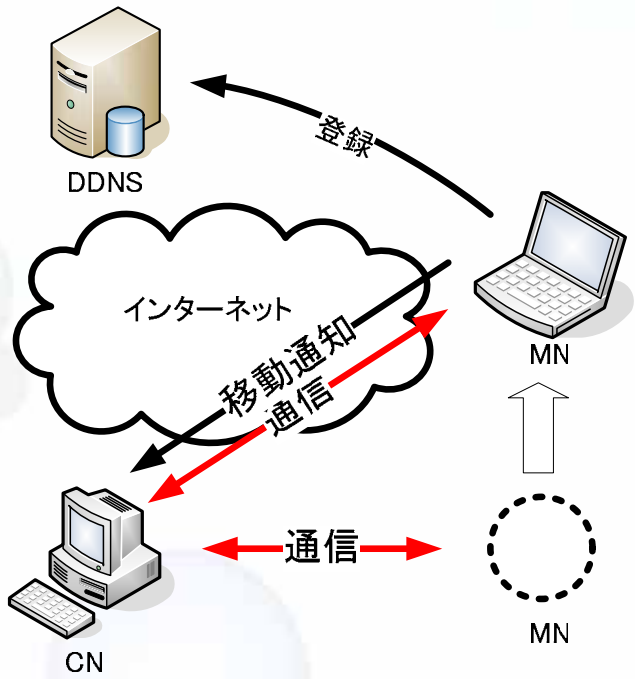
Mobile PPC (Mobile Peer to Peer Communication)

提案する通信方式

- 移動透過な通信を実現するために
  - 通信開始時において相手のIPアドレスを知る方法
    - » 初期IPアドレスの解決
  - 通信中にIPアドレスが変わった場合に通信を継続できる方法
    - » 継続IPアドレスの解決

この2つを明確に分離

- 初期IPアドレス解決には
  - ダイナミックDNSを使用
  - 移動ノードのホスト名を知っていれば通信開始可能
  - DNSの拡張で，すでに実用となっている
- 継続IPアドレス解決には
  - Mobile PPCを適用

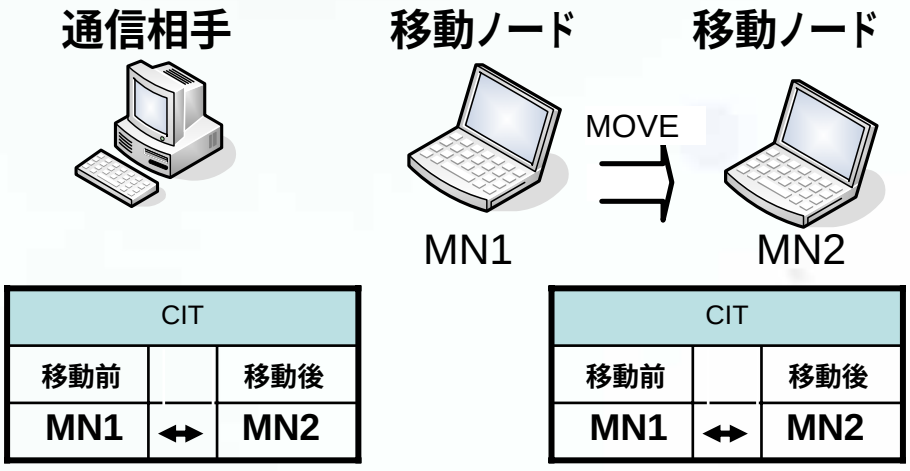


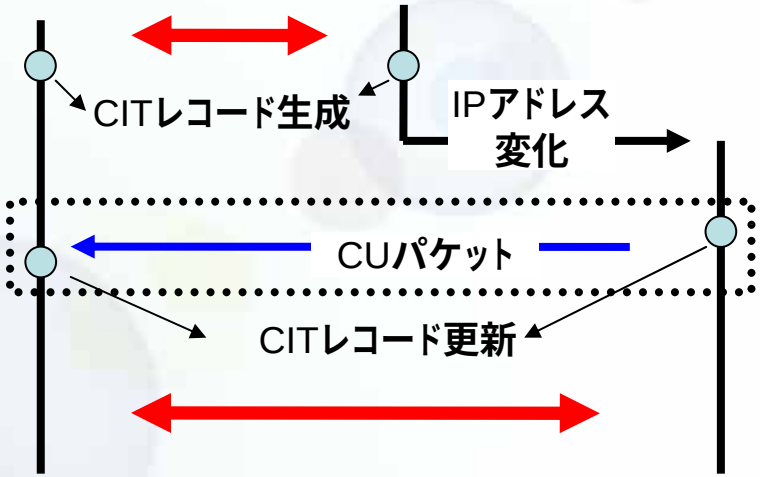
Mobile PPC

- 概要
  - エンドエンド方式でネットワーク層におけるアプローチ
  - 対応ノードは，アドレス変換に使用する移動前後の通信の対応関係を示すテーブルCIT(Connection ID Table)を保持
    - » コネクションごとに通信継続処理を適用

- エンド端末のIP層に
  - 移動時の移動通知機能
    - » 移動時にCU(CIT Update)パケットで移動情報を通知
    - » CITテーブル更新
  - IPアドレス変換機能
    - » パケット送受信時にIPアドレス変換
    - » IPアドレスの変化を上位層から隠蔽

Mobile PPCの通信 – 通信開始時 -





通信開始時

- 両端末でCITレコード登録

移動時

移動ノードの処理

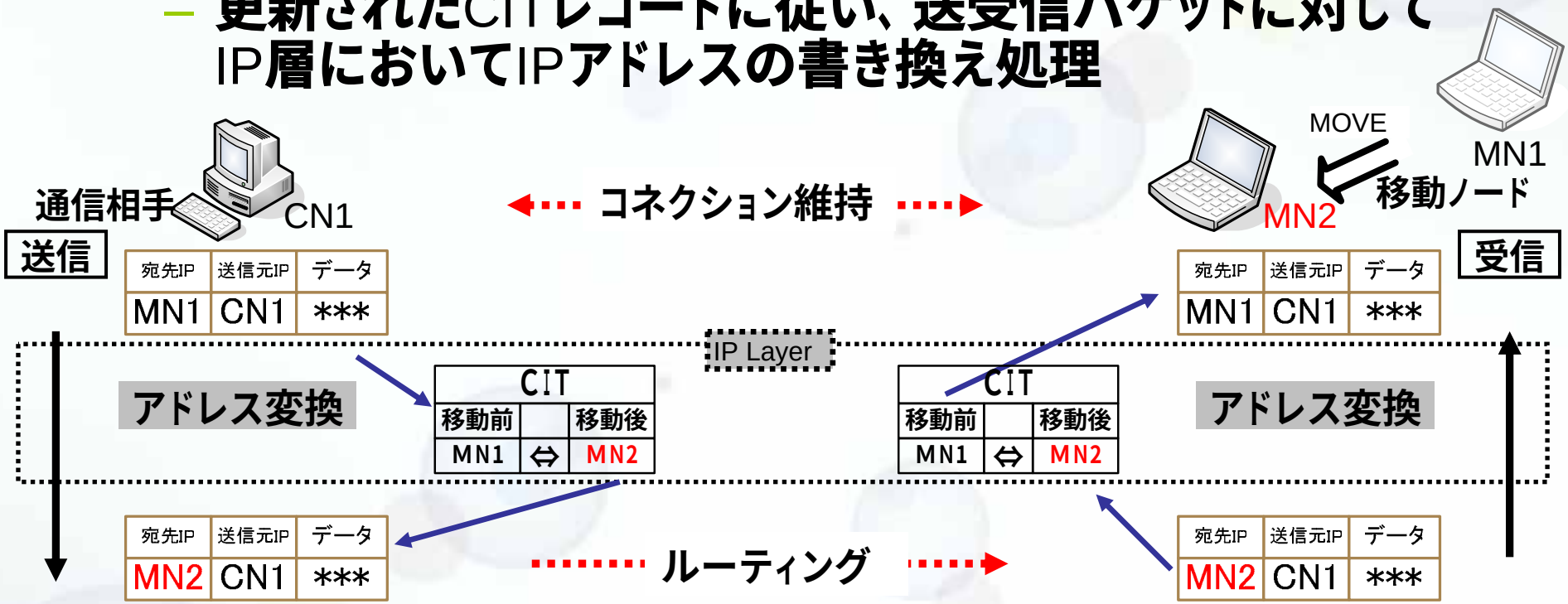
- 移動し，IPアドレスを取得
- CITレコードを更新
- CUパケットの生成･送信

通信相手ノードの処理

- CUパケットに含まれる情報を元に、CITレコードを更新

Mobile PPCの通信 – アドレス変換 -

- アドレス変換処理
  - 更新されたCITレコードに従い、送受信パケットに対して IP層においてIPアドレスの書き換え処理



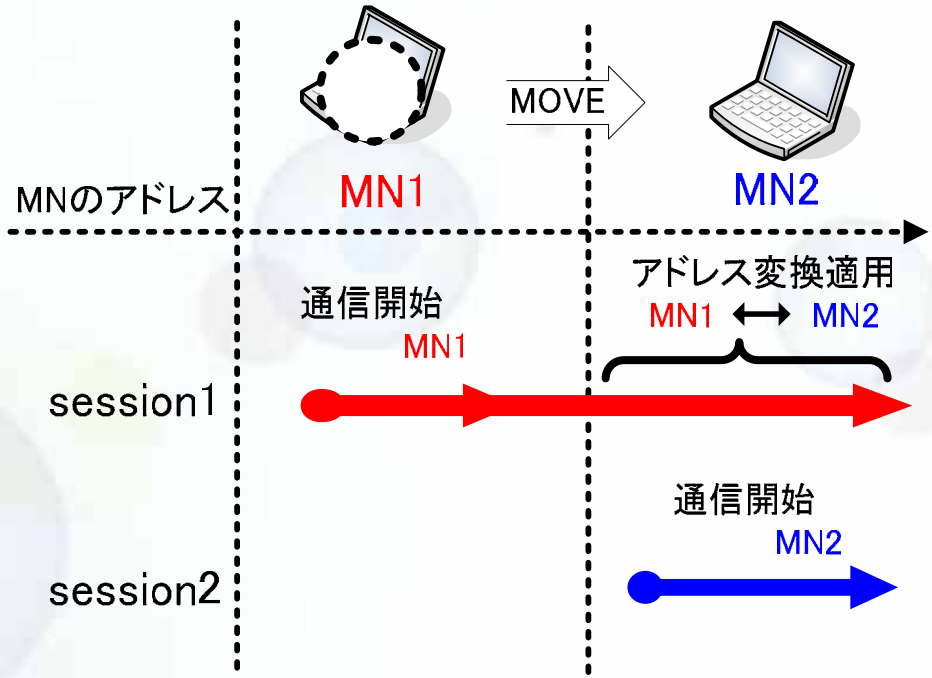
送信時
正しくルーティングされるようにアドレス変換

受信時
通信開始時のアドレスの組に一致するようにアドレス変換

IPアドレスの変化を上位層に隠蔽

アドレス変換の適用範囲

- 通信が開始されると，上位層では同じIPアドレスで通信を識別
  - IPアドレス変換処理は移動前後で継続する通信にのみ適用
    - » 通信開始時と移動後のIPアドレスによるアドレス変換
    - » 移動後に開始される通信では，新しく取得したIPアドレスが使用される



MNの上位層では

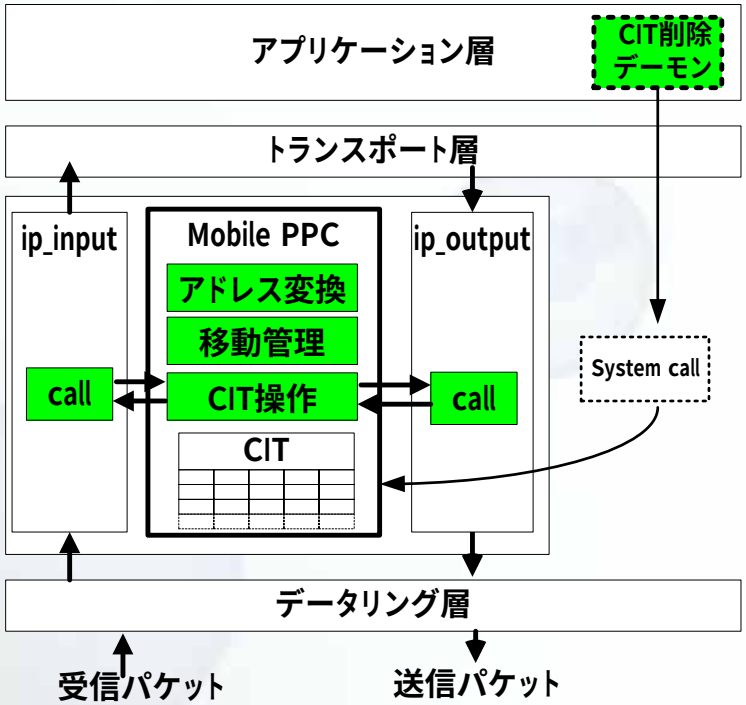
Session1はMN1

Session2はMN2

で認識している

実装

- **実装方式**
  - FreeBSD(5.2.1-R)のIP層にモジュールを組み込む
  - IP層の入出力時に呼び出し、処理を終えたら差し戻す方式
  - IP層で行われる既存の処理に変更を加えない



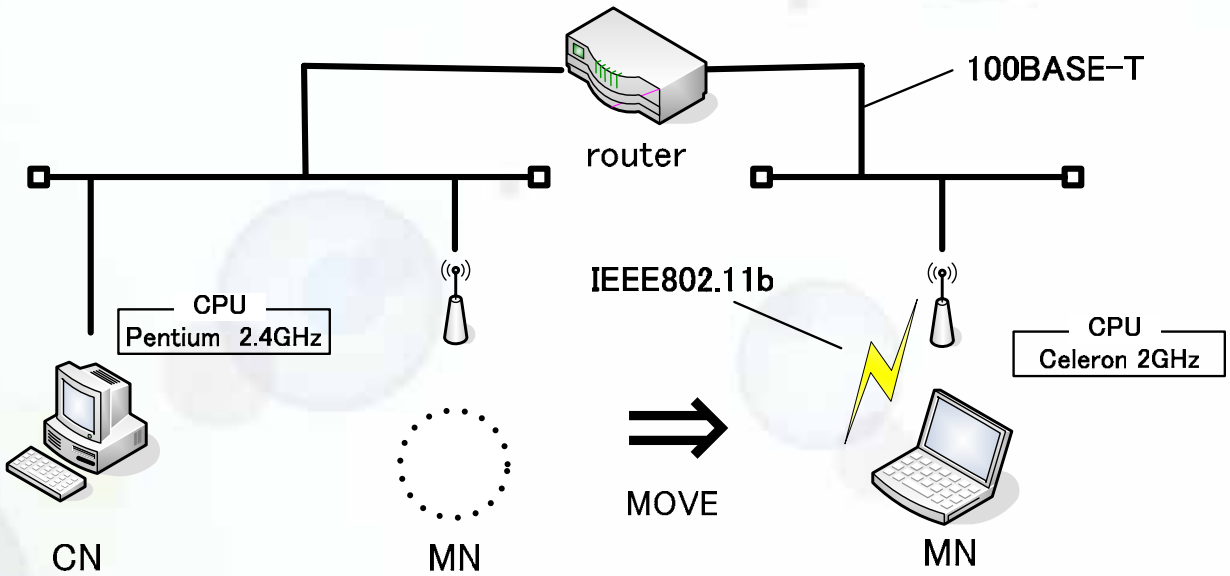
モジュール構成

アドレス変換モジュール
　IPアドレス変換，チェックサム差分計算
移動管理モジュール
　CU生成･通知
CIT操作モジュール
　CITレコードの検索･生成･更新
　（レコードの削除はデーモンが行う）

CIT：ハッシュテーブル(チェイン法)

移動透過性の検証実験

- **検証実験**
  - MNとCNにMobile PPCを実装
  - MNからCNへFTPを用いたデータ転送中に，MNのネットワークを移動させて，DHCPにより新しくIPアドレスを取得



IPアドレスは変化後も，データ通信が継続していることを確認

FTPによる性能測定

① Mobile PPCを実装していない状態
② Mobile PPCを実装し，移動前(アドレス変換なし)の状態
③ Mobile PPCを実装し，移動後(アドレス変換あり)の状態

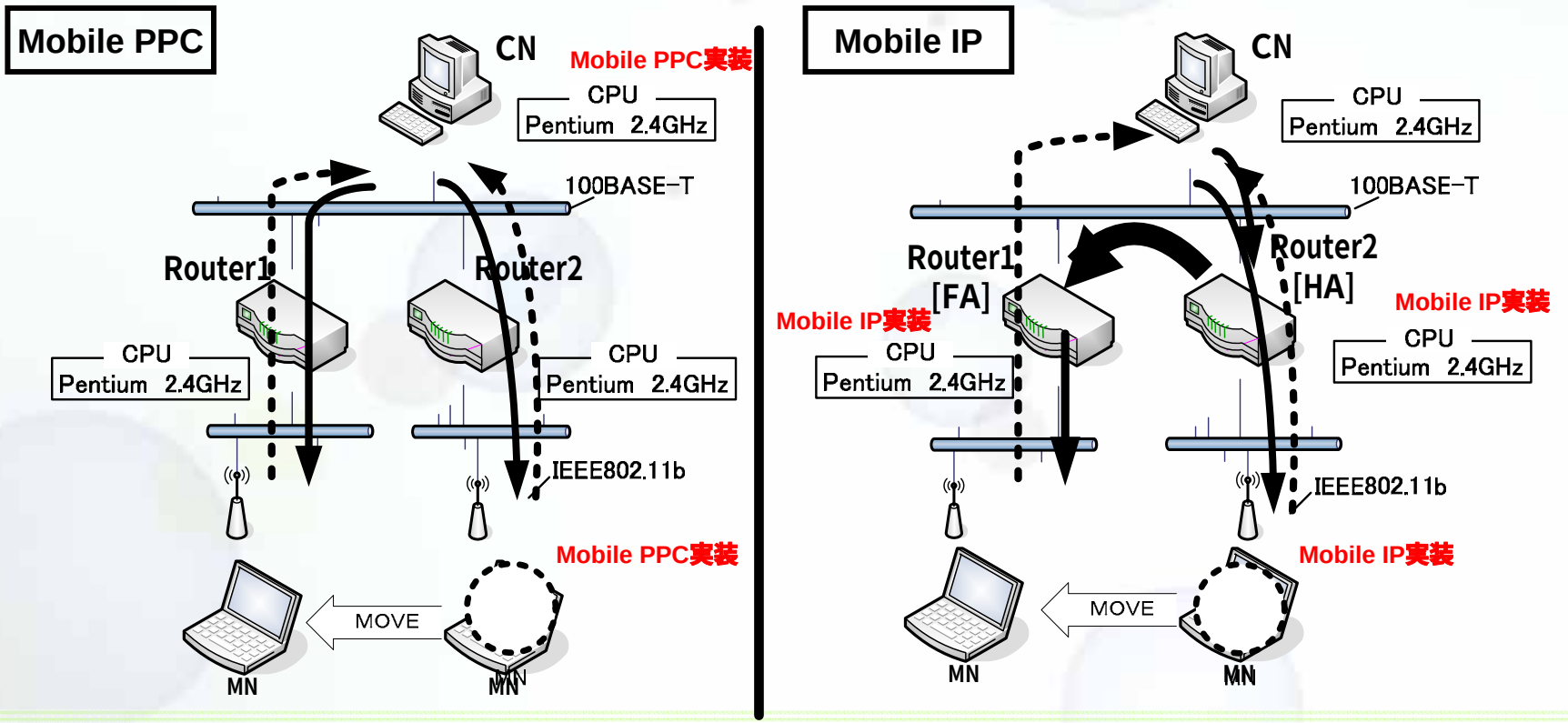
５０Mのファイルをダウンロードするのに掛かった時間を測定 (10回平均)

状態	ダウンロード時間
① 未実装	87.31 [秒]
② 移動前	87.31 [秒]
③ 移動後	87.40 [秒]

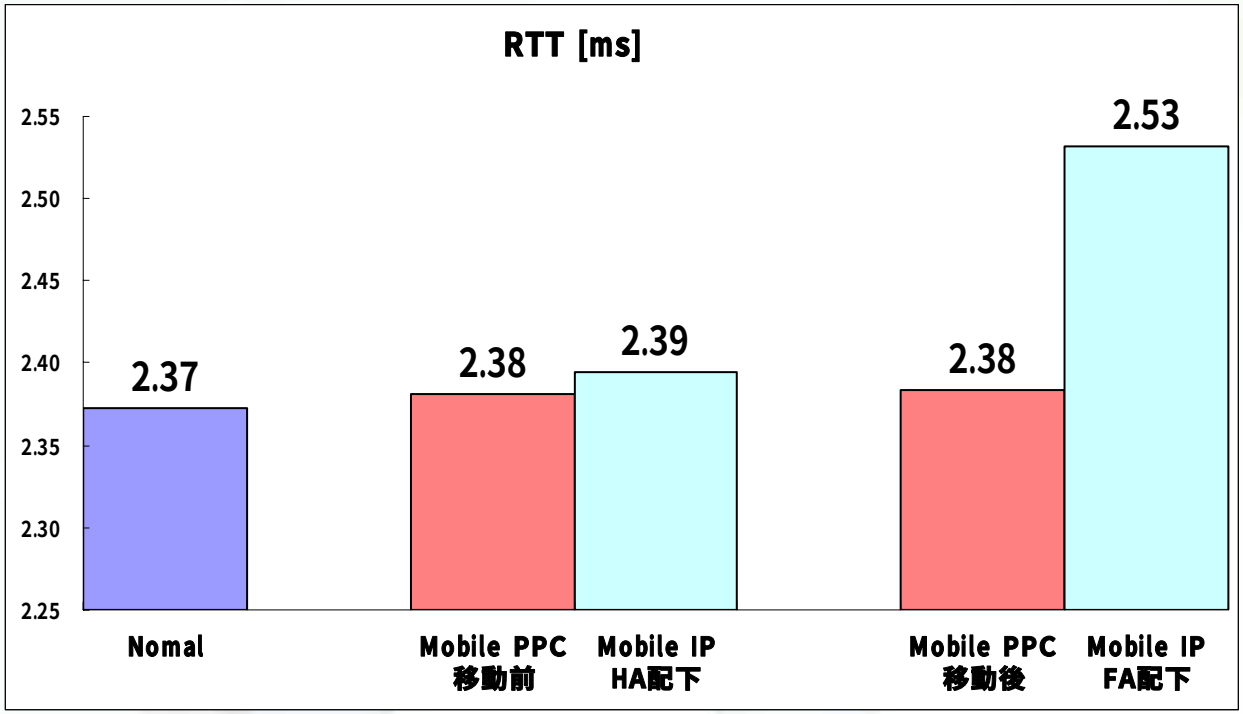
- 考察
  - 移動前のアドレス変換をしていない状態では，性能低下は見られない
  - 移動後にアドレス変換をしている状態でも，0.1%程度の処理時間増加
  - Mobile PPCによるオーバヘッドは実用上，ほとんど影響ないと考えられる

Mobile IPとの性能比較

- MN~CN間で内部処理を含めたパケット往復時間[RTT]を測定
  - MNが送信したUDPパケットをCNが受信し，CNがそのパケットをMNへ送り返す測定プログラムを試作
  - Mobile PPC，Mobile IPの移動前，移動後において測定
    - » 1000回(パケット)試行し，その平均値を算出



Mobile IPとの性能比較　結果



- Mobile PPCは移動前後で変化なし
- Mobile IPは，FA配下になると経路の冗長やトンネリングなどによるオーバヘッドの影響が出ている
- 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が実証できた

Mobile PPCの特徴

- エンドツーエンド方式のアプローチ
  - プロキシのような特殊な装置が不要
- IP層でIPアドレス変換
  - TCP/UDPを含む上位ソフトウェアを変更する必要がない
- 移動後に継続する通信は，通信開始時と移動後のIPアドレスによるIPアドレス変換のみ
  - パケット長の変化はなく，通信経路の冗長も生じない
- 既存の処理に変更を加えない実装方式
  - 性能の低下はほとんどない

課題

- Mobile PPCにおける課題
  1. 移動時の認証機能
     - エンド端末間であらかじめ保持した共有鍵での認証を行っている
     - グローバルな環境で使用できる認証機能の検討
  2. 移動時のパケットロス
     - IPv4環境では，MNが単独でネットワークの移動を検知するのが困難であり，DHCPなどからのIPアドレス取得に時間がかかる
     - 無線レイヤとの連携によるより高速なハンドオーバー処理が必要
  3. 移動ノードの同時移動
     - 移動ノード同士が全く同時に移動した場合にCUメッセージが伝わらず通信が継続できなくなる可能性がある
     - 一時的に，新旧IPアドレスの両方を保持するような対策が必要

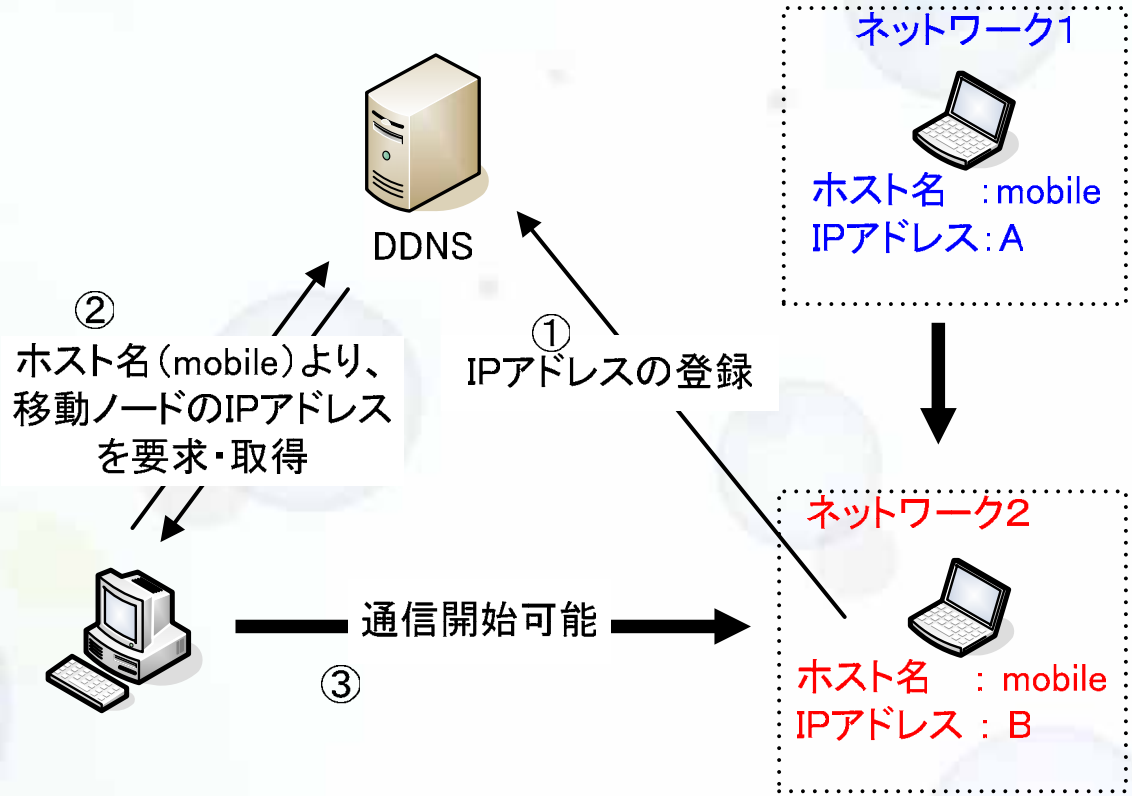
むすび

- まとめ
  - Mobile PPCによる移動透過性を提案した
    - » エンドツーエンド方式で特殊な管理サーバが不要で導入が容易
  - Mobile PPCを実装し，性能測定を行った
    - » アドレス変換によるオーバヘッドは，実用上問題ない
    - » Mobile PPCの処理が通信に与える影響は，Mobile IPに比べて小さい
- 今後
  - 課題の解決を検討
  - IPv6への適用を進めていく

おわり

初期IPアドレスの解決

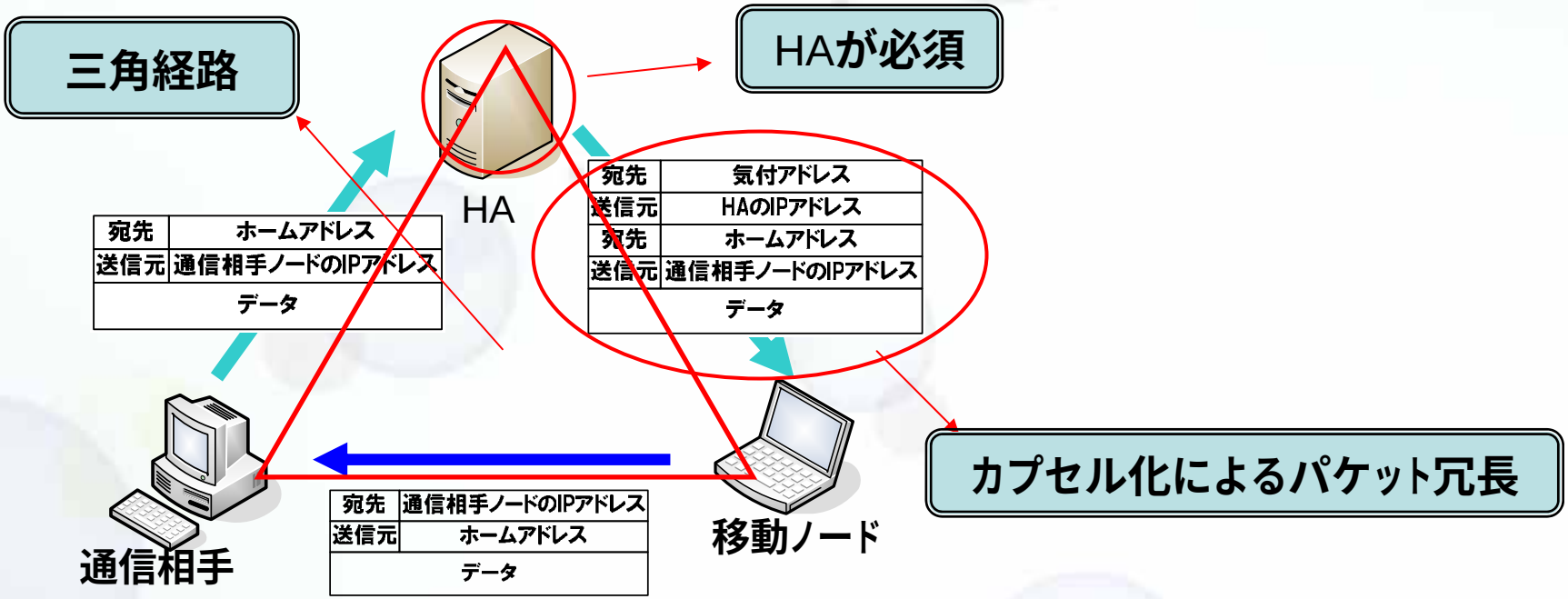
- DDNSを使用した、初期IPアドレスの解決



移動透過性を実現する技術とその課題①

- プロキシ方式
  - パケットをプロキシが中継し、移動ノードへの転送

例・・ **Mobile IP**



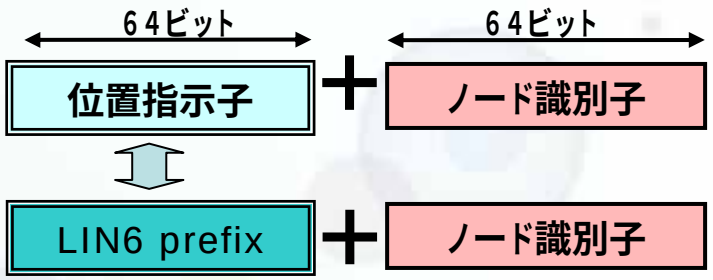
移動透過性を実現する技術とその課題②

- エンドツーエンド方式
  - 通信相手は位置管理サーバより移動ノードのアドレス取得
  - エンド端末間でIPアドレスの変化の隠蔽

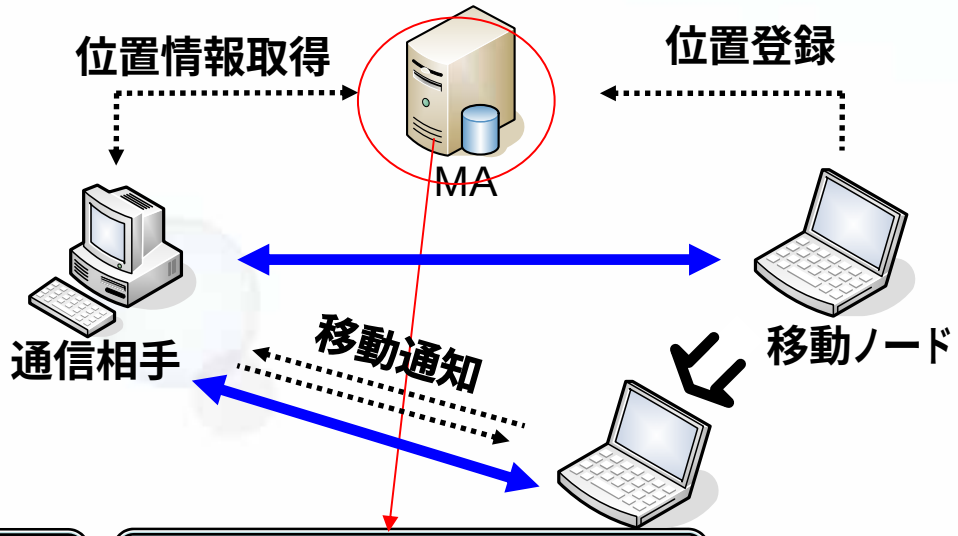
例・・

LIN6

IPv6アドレスを位置指示子とノード識別子に分離した縮退アドレスモデルを導入



位置指示子とノード識別子の対応を保持



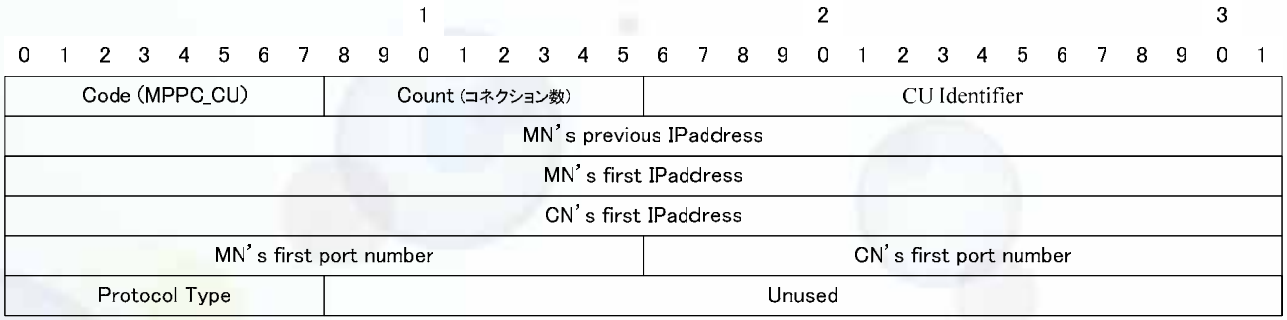
IPv4への適用は困難

アドレスの割り当てとその管理機構が必要

特殊な位置管理エージェント

CUメッセージフォーマット

- CUパケットのフォーマット
  - ICMP Echo Requestをベース
  - 通知情報として
    - » MNの移動前後のIPアドレス
    - » エンド端末間で接続している全コネクション識別子

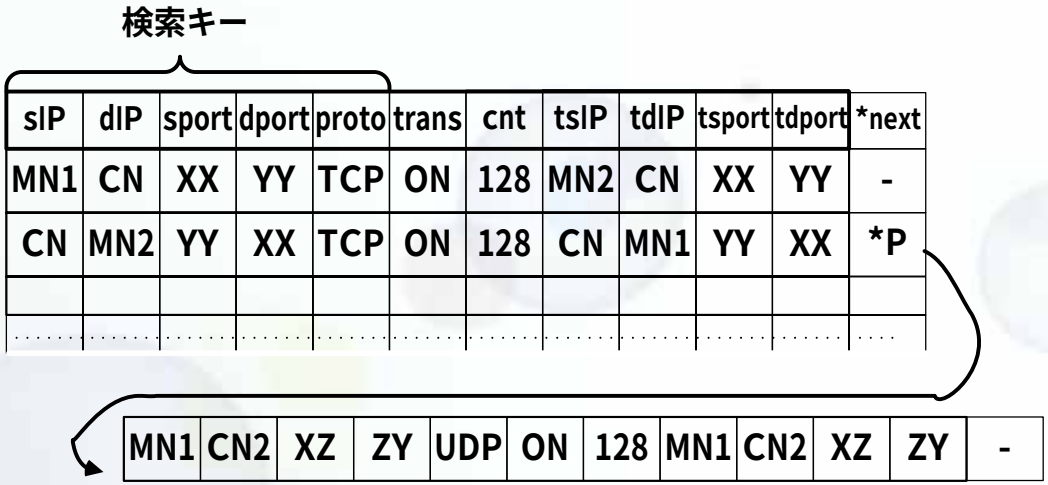


接続しているコネクションの数だけ通知情報を追加

1つの通信に対する通知情報

CITフォーマット

- CITはIP層で保持
  - ハッシュテーブルで実装 [チェイン法]
- 検索キーは，sIP/dIP/sport/dport/portoのハッシュ値
- cntはデーモンにより定期的にデクリメント，値が0になると削除



sIP/dIP：送受信IPアドレス

sport/dport：送受信ポート番号

proto：プロトコル番号

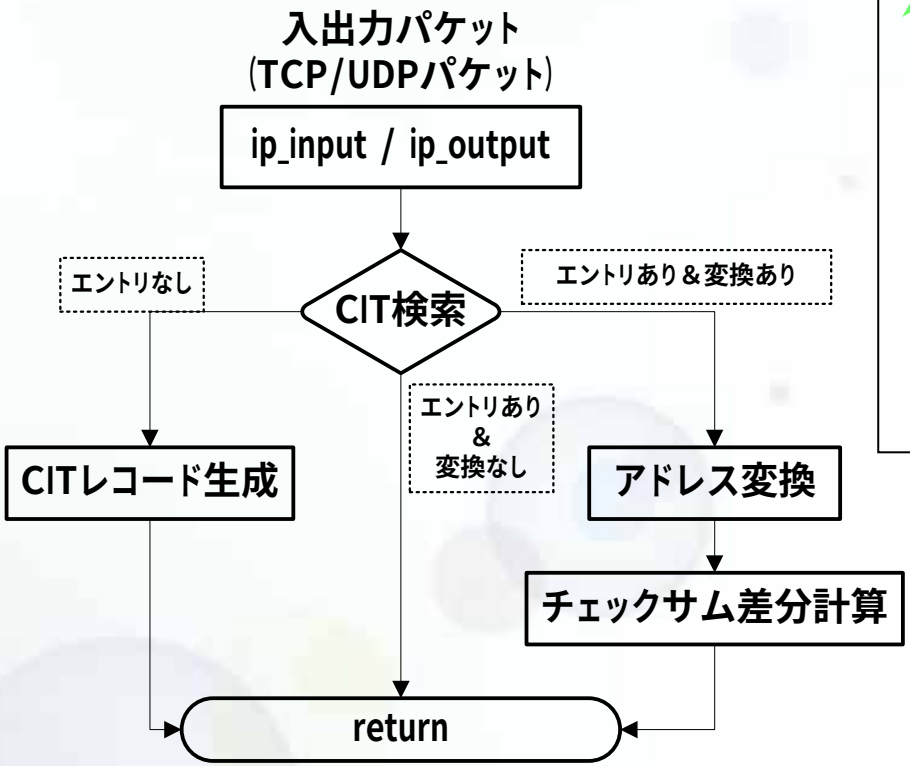
trans：変換フラグ

cnt：カウント値

tsIP/tdIP：変換先・IPアドレス

tsport/tdport：変換先・ポート番号

モジュール機能 [アドレス変換モジュール]



- IP層で入出力パケットを監視
- コネクション識別子を元にCIT検索
  - にエントリなし
    - » CITレコード生成
  - エントリあり＆変換先なし
    - » 差し戻す
  - エントリあり＆変換先あり
    - » アドレス変換処理を適用

- 除外パケット
  - DNSパケット
  - DHCPパケット
  - RIPなどの経路制御御パケット

通信中断時間の測定

通信中断時間はIPアドレス取得時間と通知処理時間の和

- MNが移動し，通信継続するまでの時間を測定
  - DHCPによるIPアドレス取得時間 (10回試行)

最大	平均	最小
9.01[秒]	6.33[秒]	4.11[秒]

  - 通知処理時間
    - » MN・CNのCIT更新時間，BUパケットの伝達時間の合計
    - » エンド端末のコネクション数が1～6個時を測定 (10回平均)

コネクション数	1	2	3	4	5	6
通知処理時間[ﾐﾘ秒]	4.61	4.57	4.57	4.72	5.08	5.49

通信中断時間の大半は，IPアドレスの取得にかかる時間

未実装端末との通信

- Mobile PPCでは
  - 通常のIPv4アドレスを使用
  - 移動通知に用いるCUはICMP Echo Request (Ping) をベース

Mobile PPCを実装していない一般端末とも通常どおりの通信が可能

- Mobile PPC対応端末と一般端末が通信中に移動
  - 移動透過な通信のサポートはできない
  - 一般端末は，DDNSによりMobile PPC対応端末のIPアドレスを知ることができる

移動後に，一般端末または移動端末から通信を開始することは可能