

ネットワーク単位の移動透過性を実現する Mobile NPC の実装と評価

坂本 順一[†] 鈴木 秀和[†] 渡邊 晃[†]

名城大学大学院理工学研究科[†]

Implementation and its evaluation of Mobile NPC realizing Network Mobility

Junichi Sakamoto Hidekazu Suzuki Akira Watanabe

Graduate School of Science and Technology, Meijo University

1. はじめに

無線 LAN やインターネットの急速な普及により、ユビキタス社会を実現するために移動しながらどこでも通信ができる環境が要求されている。しかし、インターネットでは通信を行っている端末が移動すると IP アドレスが変化するため通信が継続できない。そこで、移動によって IP アドレスが変わっても通信を継続できる移動透過性の研究が行われている。また、電車内や自動車内にもネットワークを構築し、そのネットワーク自体が移動しても、ネットワーク内の端末と外部端末との通信が継続できるネットワーク単位の移動透過性の研究も行われている。

移動透過性を実現する方式はプロキシ方式とエンド・ツー・エンド方式に分類される。プロキシ方式は相手通信端末からのパケットをプロキシサーバが中継し、移動ノード (Mobile Node; 以下 MN) に転送を行う方式である。エンド・ツー・エンド方式はプロキシサーバを利用せずにエンド端末間で移動透過な通信を行う方式である。

端末単位の移動透過性を実現させたプロキシ方式の既存技術として Mobile IP^{1), 2)}がある。また Mobile IP を利用してネットワーク単位の移動透過性を実現させた技術として Network Mobility (以下 NEMO)³⁾や、Mobile IPv6⁴⁾に基づいた NEMO Basic Support Protocol⁵⁾などが RFC で定義されている。3) 5) は Mobile IP の技術を利用しているため、Home Agent (以下 HA) のような特殊なサーバの設置が必要なことや、HA を介して通信を行うための通信経路の冗長や、トンネル化によるヘッダオーバーヘッドなどの課題がある。

我々は端末単位の移動透過性をエンド・ツー・エンド方式で実現させた Mobile Peer to Peer Communication (以下 Mobile PPC)⁶⁾を提案している。Mobile PPC は MN の移動前後の情報を記憶しておき、IP 層でアドレス変換することにより IP 層より上位層に対してアドレスの変化を隠蔽させてコネクションを維持することができる。

本稿では、Mobile PPC と NATP を組み合わせることにより、ネットワーク単位の移動透過性を実現する Mobile Network to Peer Communication (以下 Mobile NPC) について検討した結果を示す。Mobile NPC は、特殊な装置の設置が不要でネットワーク内のアドレスはプライベートアドレスでよいという特徴がある。Mobile NPC を実装し、評価した結果、移動端末の集団をネットワークとして集約した効果を示すことができた。

以下 2 章で NEMO の課題、3 章で Mobile PPC、4 章で Mobile NPC、5 章で実装、6 章で評価を行い、7 章でまとめる。

2. NEMO の課題

NEMO は、端末単位の移動透過性を実現した Mobile IP を利用してネットワーク単位の移動透過性を実現する技術である。移動するネットワーク内には複数の端末が存在し、それらの端末は Mobile Router (以下 MR) を介して通信を行う。移動ネットワーク内の端末のアドレスは移動しても変わらない。また HA は、MR の移動ネットワークのネットワークプレフィックスを保持する。HA がそのネットワークプレフィックスと同じパケットを受信したとき、そのパケットを MR に転送する。

NEMO は、Mobile IP の技術を利用しているため、HA を介して通信を行うために発生する通信経路の冗長、HA と MR 間の双方向トンネル化によるヘッダオーバーヘッド、HA に障害が起けると通信できなくなる一点障害など、Mobile IP と同様の課題がある。

3. Mobile PPC

Mobile PPC は、HA のような特別なサーバが不要で、端末の移動透過性をエンド端末のみで実現する技術である。Mobile PPC では、移動端末 (MN) の移動前後の情報を両エンド端末が IP 層で記憶しておき、IP 層でアドレス変換することにより上位層に影響を与えないまま移動透過性を実現することができる。

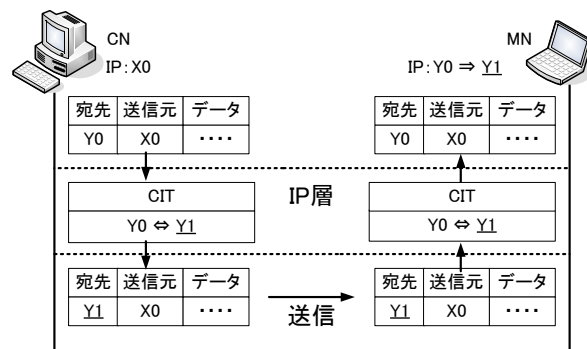


図 1 Mobile PPC のアドレス変換処理

以下に Mobile PPC の動作を説明する。ここで CN と MN は Dynamic DNS⁷⁾を用いて既に通信が開始されているものとする。MN が通信中に移動して、IP アドレスが Y0 から Y1 になると、MN は CN に対し、移動の通知(CIT Update : 以下 CU)を送信する。CN は CU を受信すると自らのアドレス変換テーブル (Connection ID Table ; 以下 CIT) を更新して MN へ CU 応答を送信する。MN は CU 応答を受信すると CN と同様に自身が IP 層に保持する CIT を更新する。

図 1 に CIT の更新後のアドレス変換処理を示す。図中の CIT は簡略化してあり、変換すべき情報のみを記述している。CN が MN へパケットを送信するときは、IP 層で CIT を参照して、宛先アドレスを Y0 から Y1 に変換して送信する。パケットを受信した MN は IP 層で CIT を参照して、宛先アドレスを Y1 から Y0 に変換して上位層へ渡す。MN が CN にパケットを送信する場合は上記と逆の変換処理を行う。

このように MN が通信中に移動しても、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、上位ソフトウェアに対してはアドレスの変化を隠蔽することにより、コネクションを維持することができる。

4. Mobile NPC

4.1. 概要

Mobile PPC は、IP 層でアドレス変換させることにより、上位ソフトウェアに対してアドレスの変化を隠蔽することができる。Mobile NPC はこの特徴を活かしたもので、IP 層より上位層で動作する NAPT に対してアドレスの変化を隠蔽させることによりネットワーク単位の移動透過性を実現させる。

Mobile NPC の移動ネットワークの構成を図 2 に示す。移動ネットワークは、Mobile PPC と NAPT を実装した Mobile NPC Router(以下 MPR)によりインターネットと接続される。移動ネットワーク内は IPv4 のプライベートアドレス空間とし、複数の一般端末 Private address Terminal (以下 PT) が存在できる。プライベートアドレスでよいから、移動ネットワーク内のアドレスを自由に割り当てることができ、アドレス管理が容易である。MPR が NAPT 機能を内蔵するため通信に制約があるが、NAT 越え技術と併用することによりこの制約を除去する。NAT 越え技術については 4.3 で考察する。

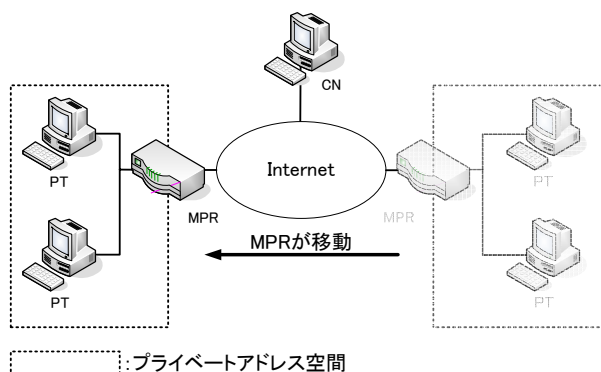


図 2 移動ネットワークの構成図

4.2. Mobile NPC における移動透過性の実現

ネットワーク移動前、移動時、移動後の通信処理を以下に示す。

(1) 移動前の処理

PT が CN に通信を開始する場合の処理を図 3 に示す。CN の X0 と MPR の Y0 はグローバルアドレス、PT の P0 はプライベートアドレスである。移動ネットワーク内の PT が CN に向けて最初のパケットを送信する。上記のパケットを受信した MPR は NAPT テーブルを生成する。その後、この内容に基づきパケットの送信元アドレスを P0 から Y0 に変換して IP 層に渡す。次に MPR の IP 層では、Mobile PPC により CIT を作成してからパケットを送信する。

CN は受信したパケットから CIT を作成してから上位層へ渡す。この時点で、CIT にはアドレス変換情報は書かれておらず、Mobile PPC によるアドレス変換の処理は行われず、以後の通信では NAPT によるアドレス変換の処理のみが行われ、CN と PT の通信が行われる。

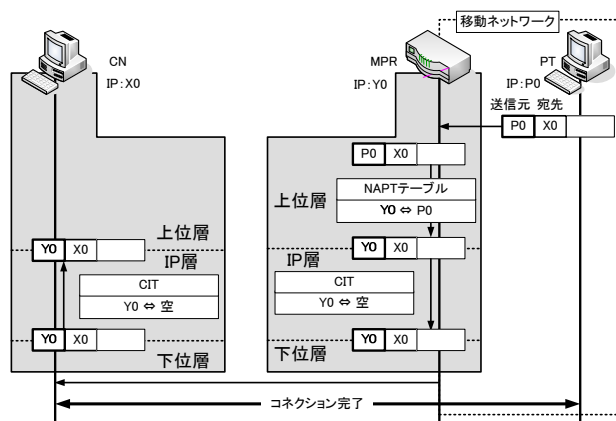


図 3 PT と CN の通信開始処理

(2) 移動時の処理

移動ネットワークが通信中に移動して MPR のグローバルアドレスが Y0 から Y1 になると、Mobile PPC の手順に従い CIT を更新する。このとき NAPT テーブルにはいっさい影響がない。CIT の内容は、Y0 と Y1 を変換する旨記述される。

(3) 移動後の処理

移動後の処理を図 4 に示す。例として PT から CN へパケットを送信した場合を示す。PT からのパケットを受信した MPR は NAPT テーブルを参照して、送信元アドレスを P0 から Y0 に変換し、MPR の IP 層へ渡す。IP 層では Mobile PPC

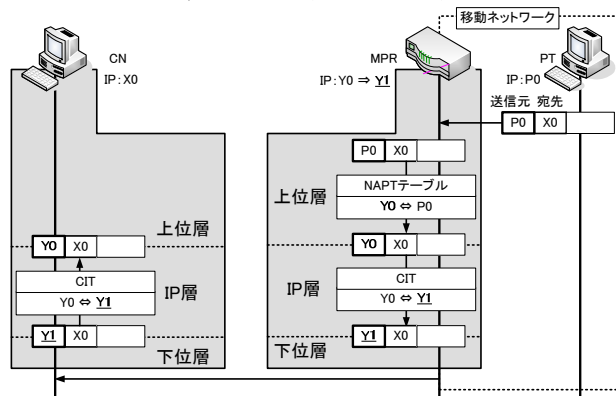


図 4 移動後の通信処理 (PT ⇒ CN)

により CIT を参照して、送信元アドレスを MPR の移動前のアドレス Y0 から移動後のアドレス Y1 に変換して、CN へ送信する。CN は受信したパケットを IP 層で Mobile PPC により CIT を参照して、送信元アドレスを Y1 から Y0 に変換して、上位層へ渡す。逆方向のパケットは上記と逆の変換を行う。

このようにして PT と CN が通信中にネットワークが移動しても、CN の上位ソフトウェアと MPR の NAT はアドレスの変化に気づかず、コネクションを維持することができる。

4.3. NAT 越えについて

既存の NAT 越えの技術として、STUN⁸⁾、UPnP⁹⁾、NAT-f¹⁰⁾ がある。STUN はあらかじめプライベート空間側の端末がインターネット上に公開された STUN サーバに利用可能なポートと NAT のグローバル IP アドレスを登録し、グローバル空間側の端末が通信開始時に STUN サーバにこの内容を問合せ。UPnP は NAT の内側の端末から NAT を制御する。NAT-f は端末と NAT が協調して NAT テーブルを強制的に生成させ、かつ端末側がポート番号の変換を行う。STUN は第三の装置を必要とし、P2P 通信を目指す Mobile NPC とは相容れない考えである。UPnP はあらかじめ内部から外部へのリンクを確立しておく必要があり、利用上の制約がある。NAT-f はエンドエンドで NAT 越えを実現し利用上の制約もない。そこで、NAT-f と Mobile NPC の融合を図るべく検討中である。

5. Mobile NPC の実装

5.1. 改造の概要

Mobile NPC を FreeBSD 上に実装し動作検証を行った。すでに開発済みの Mobile PPC をモジュールに変更を加えずにそのまま利用した。NAPT は FreeBSD に標準にインストールされている natd を利用した。本実装は、単純に Mobile PPC と NAPT を合体させたものではなく、以下のような改造が必要である。

パケット受信時に呼び出される関数 ip_input にはグローバルアドレス側、およびプライベートアドレス側から受信するパケットの両方が Mobile PPC のモジュールに渡されてしまう。パケット送信時に呼び出される関数 ip_output も同様である。そこで Mobile PPC に変更を加えず、入力・出力パケットのアドレスがグローバルアドレスの場合のみ Mobile PPC を呼び出すように改造した。

natd の場合、アドレス変換時にグローバル側のインターフェースに割り当てられるアドレスを取得してアドレス変換を行っており、アドレスが変化すると新しいアドレスを用いてアドレス変換をしてしまう。そこでアドレスが変化しても通信中のパケットは古いアドレスでアドレス変換するように natd を改造した。

5.2. モジュール構成

Mobile NPC のモジュール構成を図 5 に示す。既存の処理に変更を与えないようにパケット受信時およびパケット送信時に入力パケット判定モジュール、または出力パケット判定モジュールを呼び出し、パケットごとに Mobile PPC を呼び出すか否かを判定を行うように改造した。

CIT の生成・参照・更新、移動後のアドレス変換の処理は、Mobile PPC の処理をそのまま用いる。natd はソケットバッファを介してアドレス変換を行う。

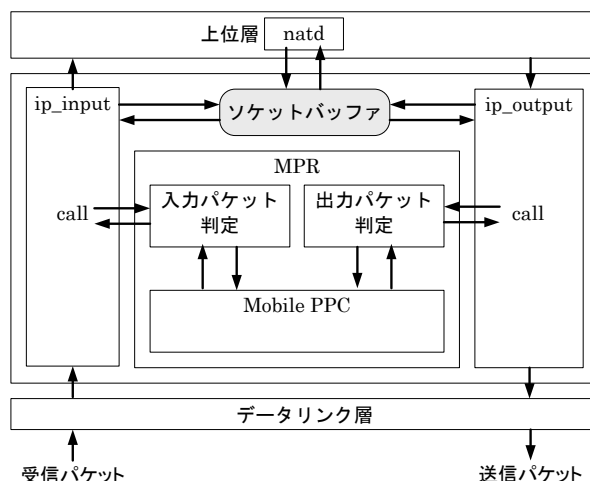


図 5 モジュール構成

5.3. natd の改造

natd ではグローバル側のインターフェースに割り当てられているアドレスを常に監視しており、そのアドレスが新しいアドレスに変わると保持しているアドレス変換テーブルをすべて削除するので、これを削除しないように改造した。

また、natd はプライベートアドレスからグローバルアドレスに変換した後、移動後のグローバルアドレスでチェックサムの差分計算を行う。しかし、変換されたグローバルアドレスは移動前のアドレスであるためチェックサムの値が異なり破棄される問題が発生する。そこで natd がアドレス変換とチェックサムの計算を行った後に、チェックサムの値が正しいかチェックを行いチェックサムの値が正しくない場合は再計算するように改造した。

6. 評価

6.1. 評価環境

図 6 に示す評価環境で動作検証を行った。CN の OS は FreeBSD で Mobile PPC を実装し、CPU は Pentium 2.4GHz、メモリが 256M、NIC は 100BASE-T である。MPR の OS は FreeBSD で Mobile NPC と NAPT を実装し、CPU は Pentium 1.7GHz、メモリが 512M、グローバル側は IEEE802.11g、プライベート側の NIC は 100BASE-T である。PT の OS は Windows XP Professional で CPU は Pentium 2.8GHz、メモリが 1024M、NIC は 100BASE-T である。

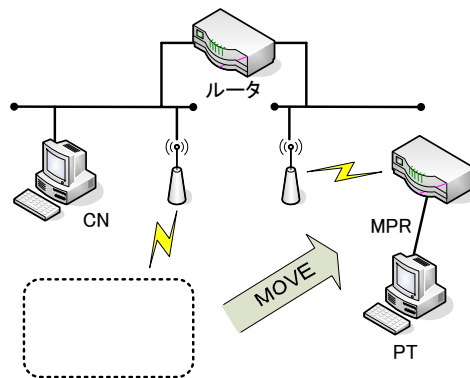


図 6 評価環境

MPR の配下の PT が、CN に対して FTP 接続を行い、ファイルをダウンロード中に MPR が移動して MPR のアドレスが変わっても CN と PT の通信を継続することができ、ファイルのダウンロードが完了することを確認した。また、PT と CN が通信中に CN が移動した場合でも同様の動作を確認した。

6.2. Mobile NPC のよる集約効果

MPR 移動時の処理を図 7 に示す。複数の PT と CN のペアが通信を行っていることを想定する。ここで、MPR が移動すると MPR は DHCP サーバから IP アドレスを取得する。次に MPR は PT と通信中の複数の CN に対し移動通知(CU)を行う。以後 MPR と CN でアドレス変換を実行することにより PT と CN の通信を継続させる。このように MPR はすべて通信中の CN に対して移動通知 (CU) を行う必要がある。Mobile NPC により集団で動く端末を集約した効果は、IP アドレス取得が MPR の 1 回だけで済むところにある。この集約の効果を示すために IP アドレス取得時間と移動通知時間 (CU) を測定した。

測定結果を表 1 に示す。IP アドレス取得時間は DHCP の 2 往復のシーケンスとその後の二重アドレスチェックを含み約 6 秒を必要とした。これに対して Mobile NPC による移動通知時間 (CU) は端末一台当り 0.4 秒以下であり、端末数を n とすると約 $0.4 \text{ 秒} \times n$ である。この結果から分かるように移動後、通信再開までの時間はほとんどが IP アドレス取得時間であり、移動通知時間 (CU) は極めて短い。移動通知がこのような短時間で実現可能なのはシーケンスがシンプルであることと処理をカーネル内で実行していることによる。以上の結果から Mobile NPC による集約効果は大きいと判断できる。

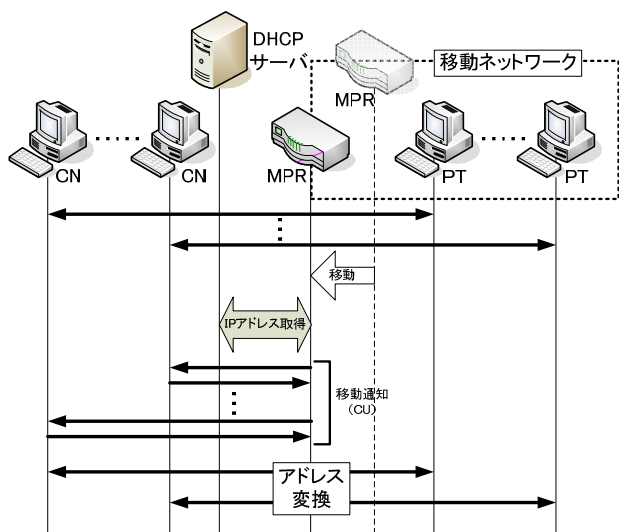


図 7 MPR 移動時の処理

表 1 測定結果

IP アドレス取得時間	6.28[秒]
移動通知時間 (CU) $\times n$	$0.373[\text{秒}] \times n$

n : 通信中の端末のペア数

6.3. FTP の性能評価

Mobile NPC を実装した場合に MPR の処理が中継性能に与える影響を調査するため PT が CN に対して FTP 接続のダウンロードにかかる時間を測定した。

MPR に (1) Mobile NPC を未実装の状態、(2) Mobile NPC を実装しアドレス変換なしの状態 (移動前)、(3) Mobile NPC を実装しアドレス変換ありの状態 (移動後) でそれぞれ PT が CN から 100M のファイルを FTP でダウンロードした時間を表 2 に示す。ケース (1) とケース (2) はほぼ同じ性能であり、ケース (3) は約 2% のオーバーヘッドが発生した。この結果により MPR が Mobile NPC 機能を保持したことによるオーバーヘッドの増加はほとんどないことが分かる。

表 2 FTP のダウンロード時間

状態	ダウンロード時間	割合
(1) Mobile NPC 未実装	129[秒]	1.00
(2) アドレス変換なし	129[秒]	1.00
(3) アドレス変換あり	131[秒]	1.02

7. むすび

本研究では Mobile PPC と NAPT を実装させた MPR を用いることによりネットワーク単位の移動透過性を実現できることを示し、その実装方法と評価結果を述べた。Mobile NPC は特殊なサーバを必要とせず、常に最適な通信経路で通信が行われ、ヘッダオーバーヘッドが発生しない。ネットワーク移動時の性能を測定し、移動端末を集約効果が大きいことを示した。また、中継によるオーバーヘッドが十分小さいこと確認した。

今後は FreeBSD に NAT-f と Mobile NPC の融合の実装し検証を行う予定である。

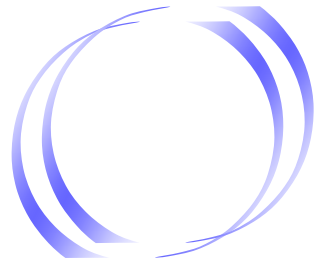
参 考 文 献

- 1) C. Perkins: IP Mobility Support, RFC2002 (1996).
- 2) C. Perkins: IP Mobility Support for IPv4, RFC3344 (2002).
- 3) T. Ernst: Network Mobility Support Goals and Requirements, Internet-Draft (2004).
- 4) D. Jhonson, C. Perkins and J. Arkko: Mobility Support in IPv6, RFC3775 (2004).
- 5) V. Devarapalli, R. Wakikawa, A. Petrescu and P. Thubert: Network Mobility (NEMO) Basic Support Protocol, RFC3963 (2005).
- 6) 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, 情報処理, DICO 2005, pp.125-128 (2005).
- 7) Vixie (Ed.), P., Thomson, S., Rekhter, Y. and J. Bound, : "Dynamic Updates in the Domain Name System", RFC 2136, April 1997.
- 8) J. Rosenberg, J. Weinberger, C. Huitema, R. Mahy, "STUN - Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)", RFC 3489 (2003).
- 9) UPnP : <http://www.upnp.org/>
- 10) 鈴木秀和, 渡邊晃: アドレス空間透過性を実現する NAT-f の実装と評価, 情報処理, DICO 2006.



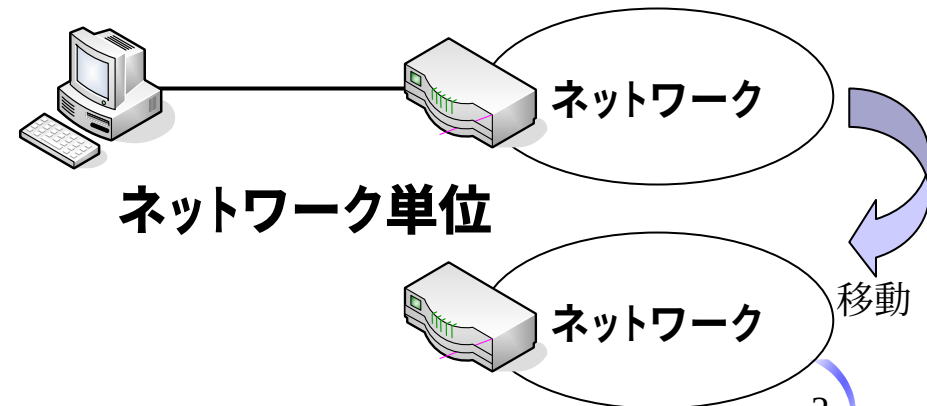
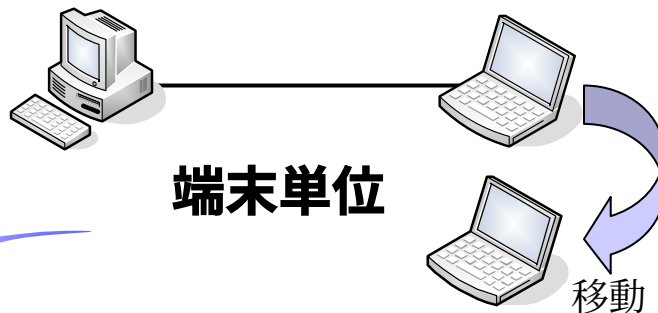
ネットワーク単位の移動透過性を 実現するMobile NPCの実装と評価

名城大学大学院 理工学研究科
坂本順一 鈴木秀和 渡邊晃



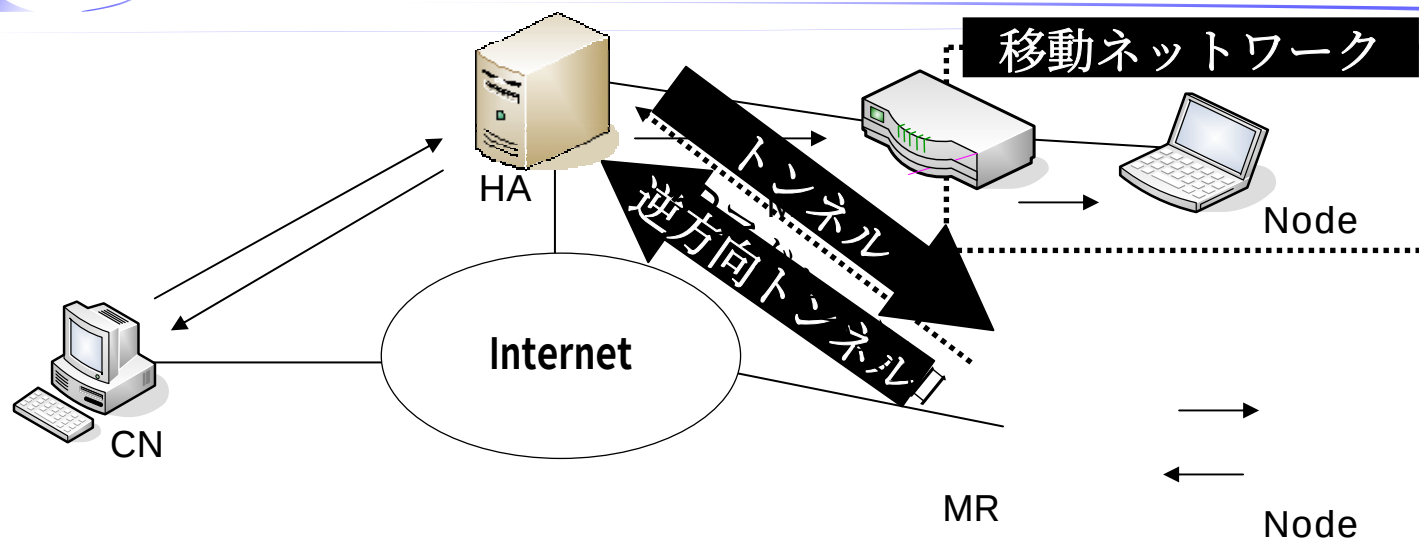
研究背景

- いつでも、どこからでもネットワークにアクセスできるユビキタス社会の実現
 - 移動しながら通信する
- 移動すると通信が継続できない
- 移動透過性の研究





NEMO(Network Mobility)の概要と課題



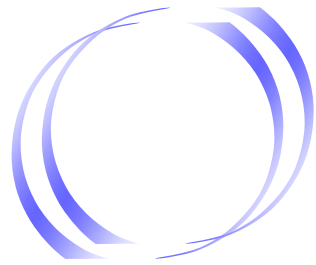
• 課題

- HAの設置
- 通信経路の冗長
- パケットのヘッダオーバーヘッド

HA : Home Agent

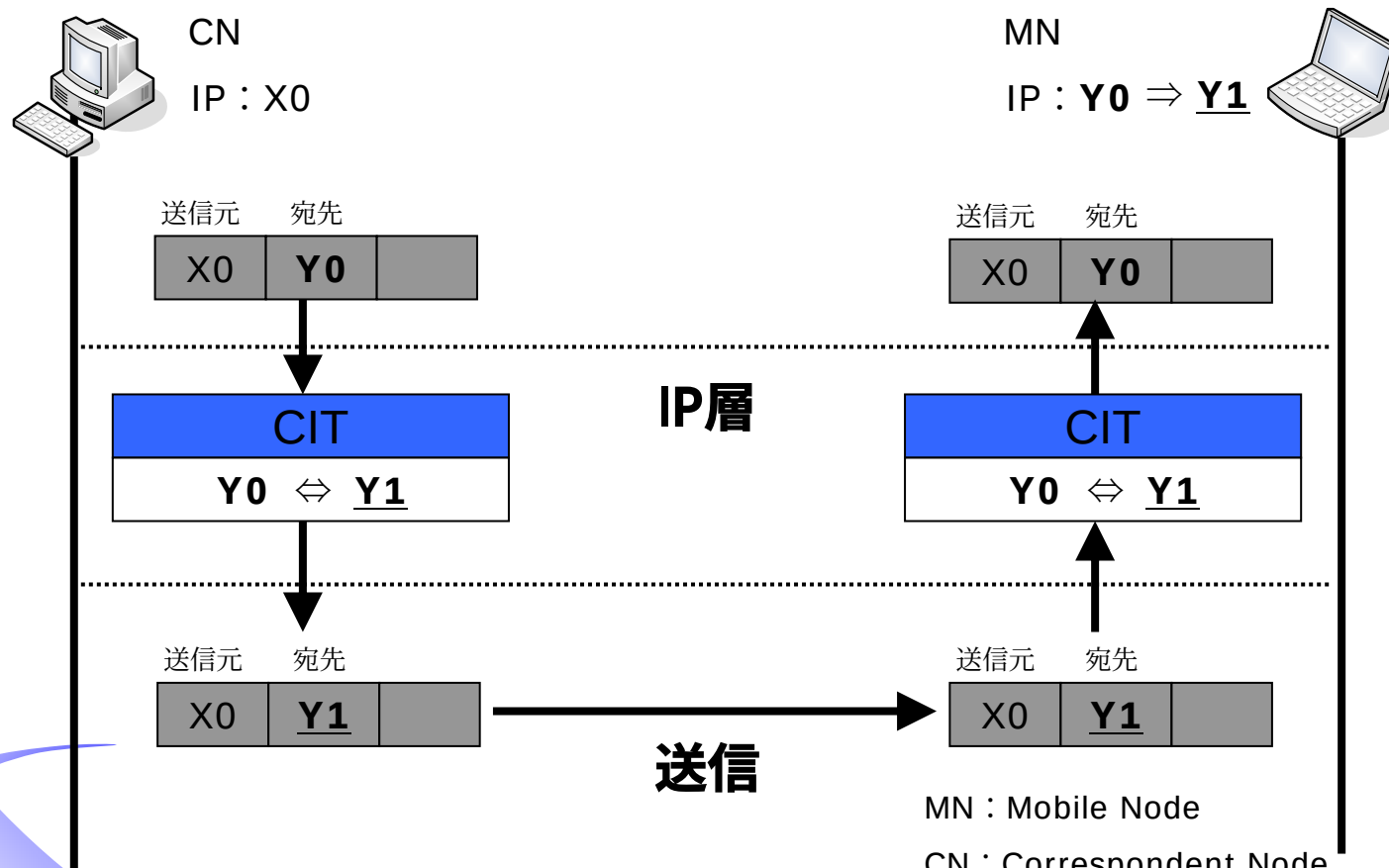
MR : Mobile Router

CN : Correspondent Node



Mobile PPC (Mobile Peer to Peer Communication)

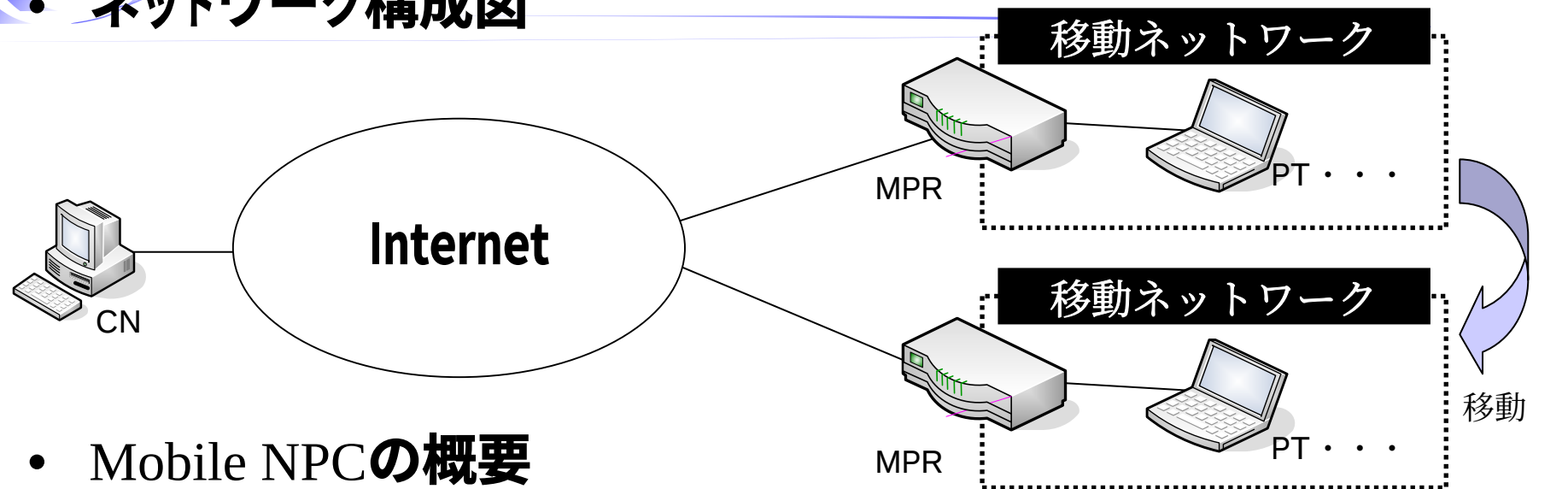
- 特別なサーバ不要、端末の移動透過性をエンド端末のみで実現



MN : Mobile Node
CN : Correspondent Node
CIT : Connection ID Table

Mobile NPC (Mobile Network to Peer Communication) の概要

- ネットワーク構成図



- Mobile NPCの概要

- 移動ネットワークはIPv4のプライベートアドレス空間
- PTは一般端末
- MPRにMobile PPCのアドレス変換とNATの機能
- CNにMobile PPCを実装

MPR : Mobile NPC Router

PT : Private address Terminal

CN : Correspondent Node

- Mobile NPCの利点

- 特殊なサーバが不要
- ヘッダオーバーヘッドなし
- アドレス管理が容易



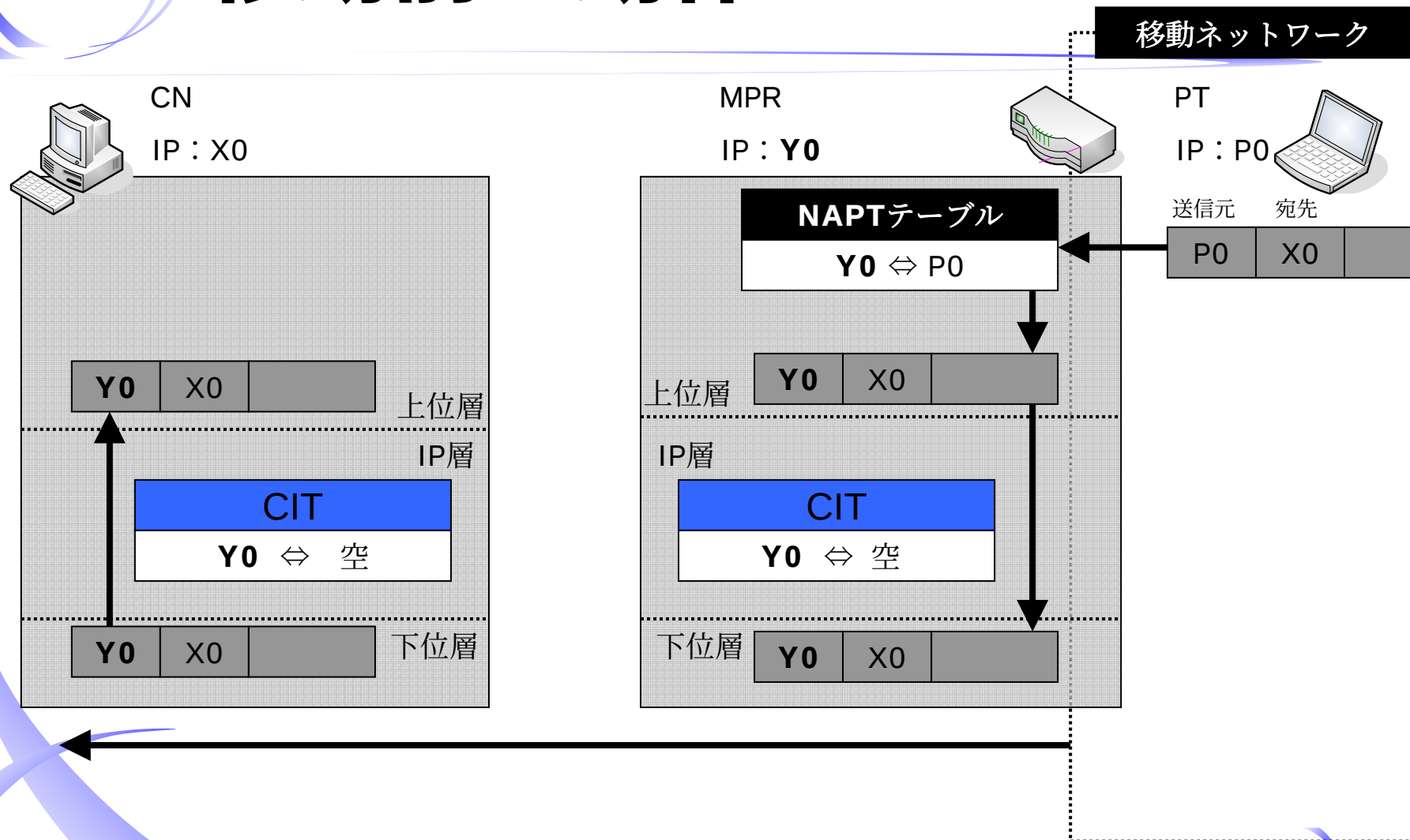
移動前の動作

MPR : Mobile NPC Router

PT : Private address Terminal

CN : Correspondent Node

CIT : Connection ID Table





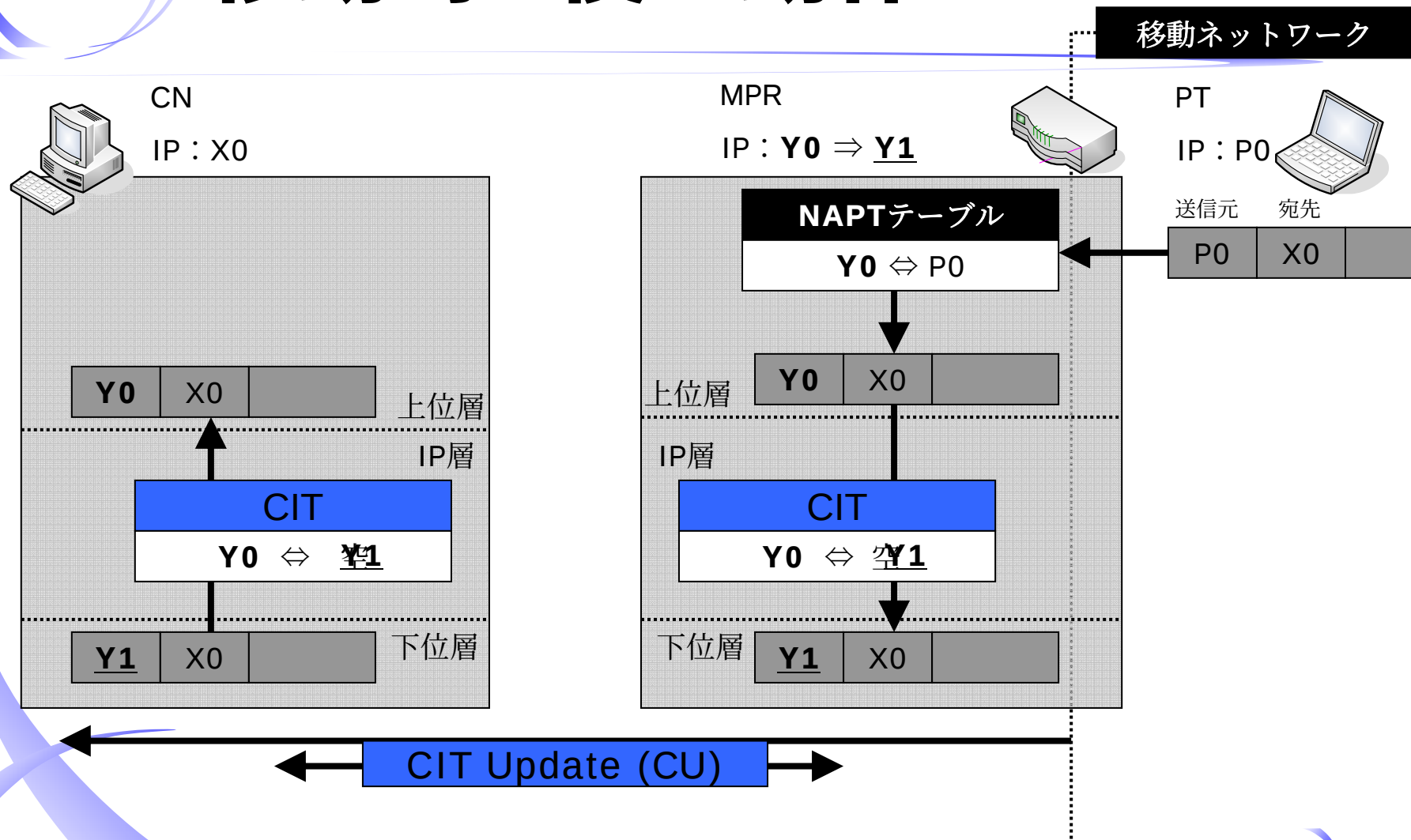
移動時と後の動作

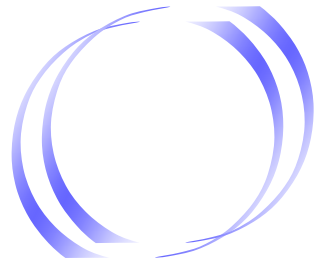
MPR : Mobile NPC Router

PT : Private address Terminal

CN : Correspondent Node

CIT : Connection ID Table



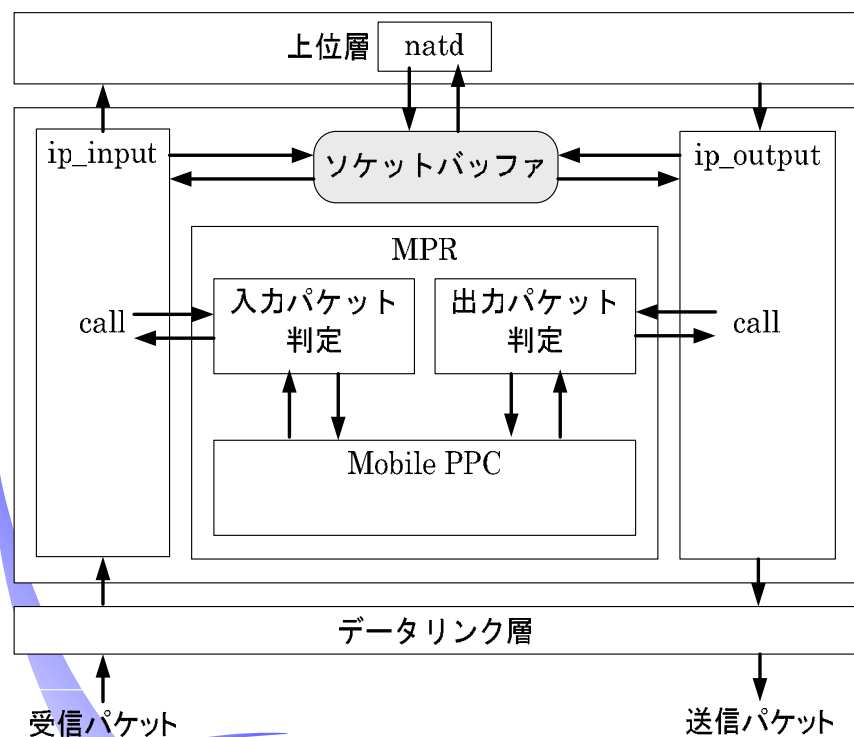


実装の概要

- OSはFreeBSD、NAPTはFreeBSDに標準にインストールされているnatdを使用
- 改造した内容
 - 呼び出し方法の変更
 - natdの改造



モジュール構成



- MPR

- 呼び出し方法の変更

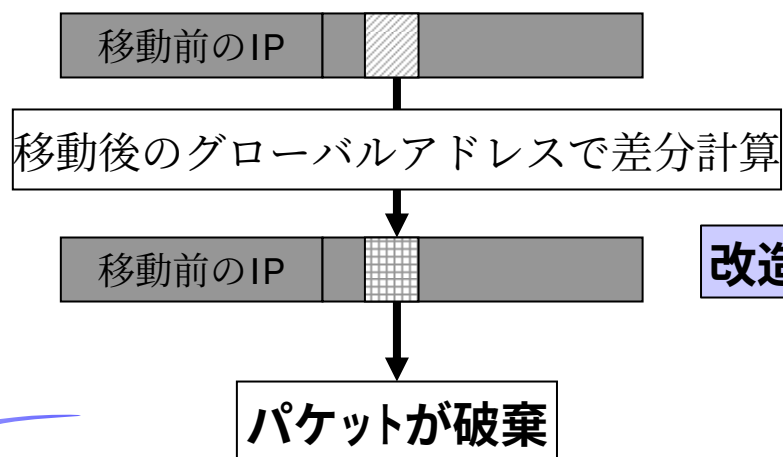
- グローバル側のインターフェースの場合のみ Mobile PPC モジュール呼び出す

- natd

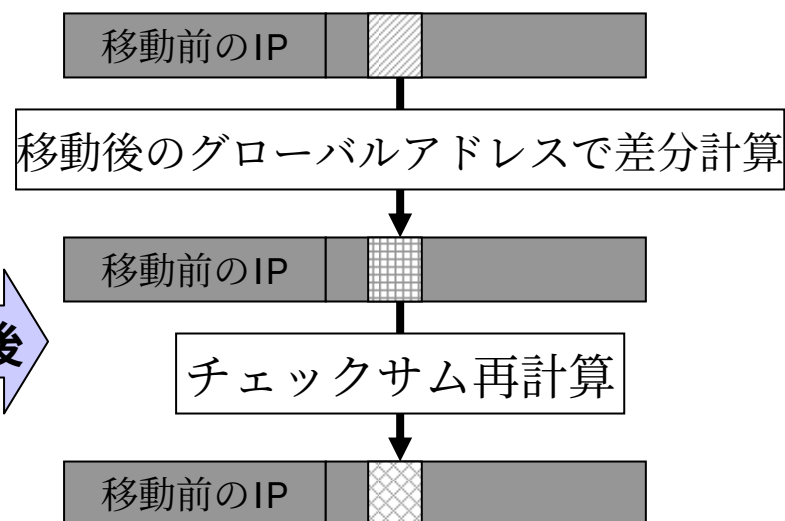
- ソケットバッファを介してアドレス変換

natdの改造

- アドレス変換テーブル
 - アドレスが変わると保持しているアドレス変換テーブルを全て削除
- ⇒削除しないように改造
- チェックサム再計算

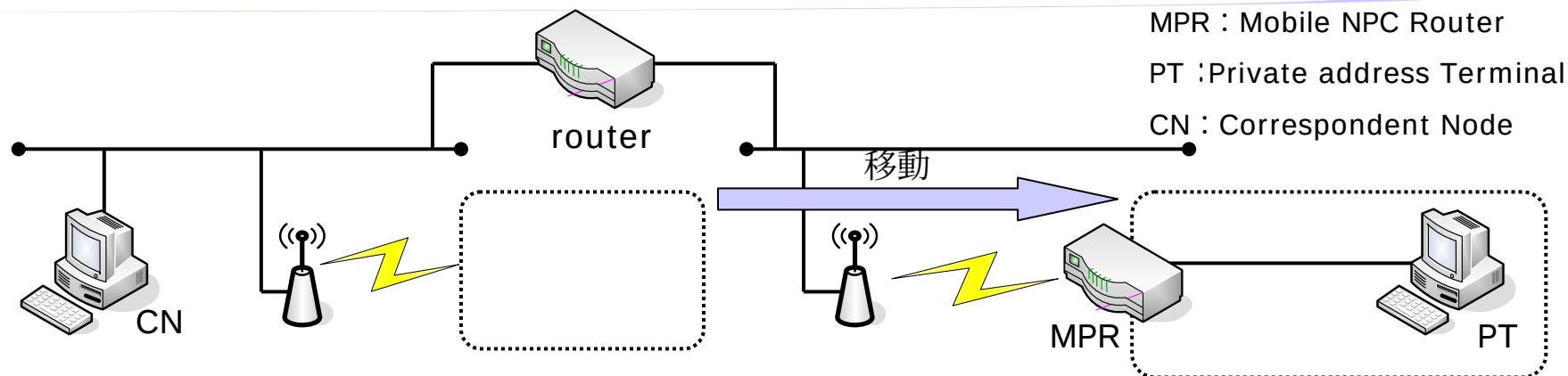


改造後





評価環境



OS:FreeBSD(Mobile PPC実装)

CPU:Pentium 2.4GHz

メモリ:256Mbyte

NIC:100BASE-T

OS:FreeBSD(Mobile NPC,natd実装)

CPU:Pentium 1.7GHz

メモリ:512Mbyte

グローバル側:IEEE802.11g

プライベート側:100BASE-T

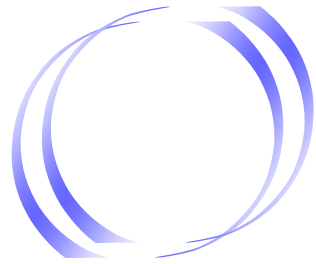
OS:WindowsXP(Pro)

CPU:Pentium 2.8GHz

メモリ:1024Mbyte

NIC:100BASE-T

- PTがCNにFTPでファイルをダウンロード中にMPRを移動しMPRのアドレス変わっても、ファイルのダウンロードの完了を確認
- CNが移動した場合も同様の動作を確認



評価内容

- **通信切断時間**
 - 移動により、通信が切断されてから再開されるまでの時間
- **FTPの性能評価**
 - Mobile NPCを実装したMPRの処理による中継性能の影響
 - PTがCNに対してFTP接続によるファイルのダウンロード時間



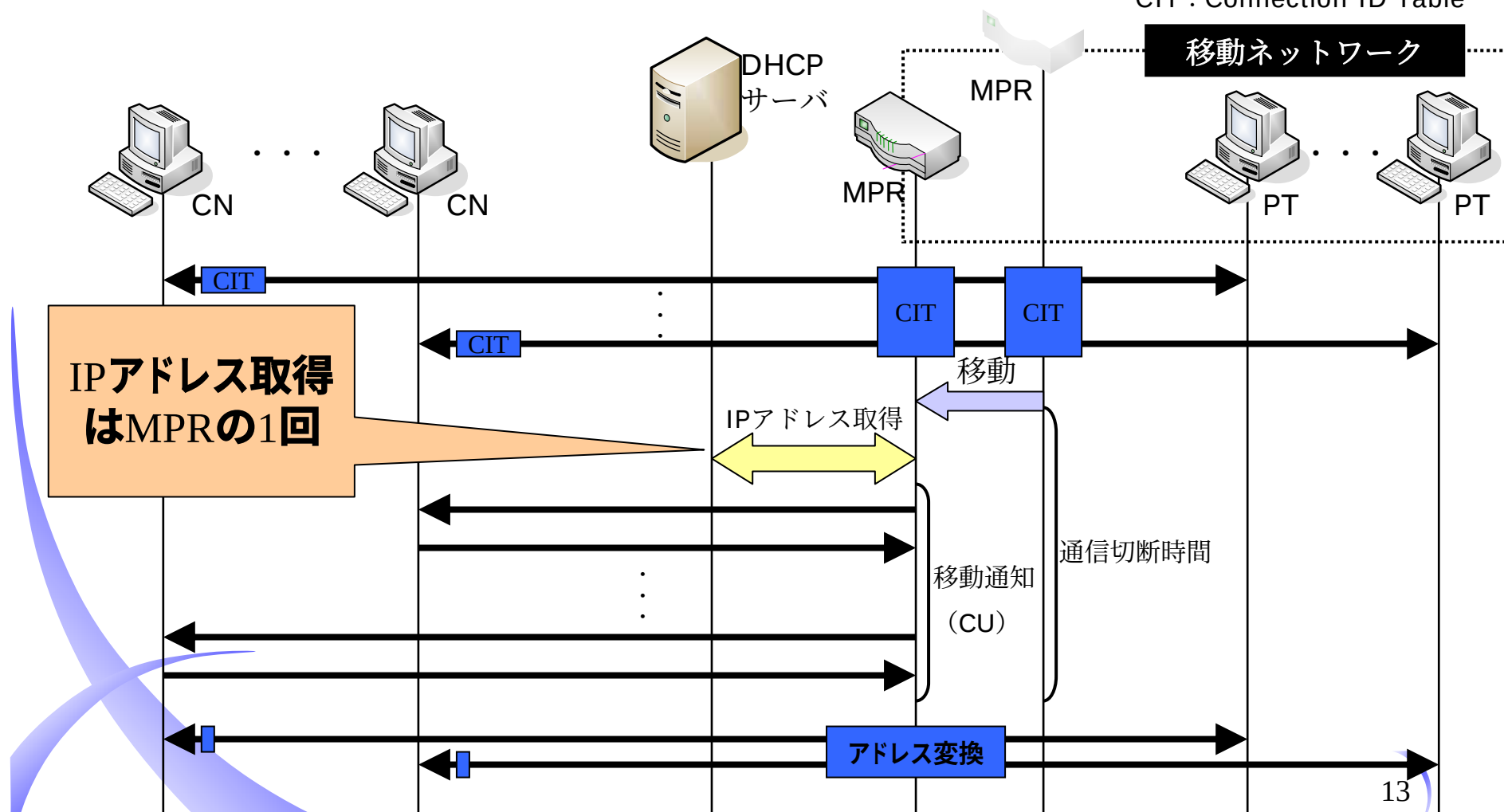
通信切断時間について

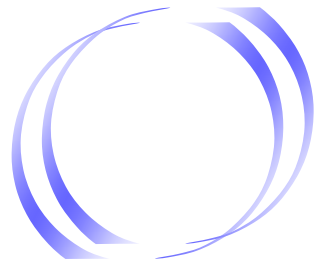
MPR : Mobile NPC Router

PT : Private address Terminal

CN : Correspondent Node

CIT : Connection ID Table





評価① (通信切断時間)

IPアドレス取得時間	6.28[秒]
移動通知時間 (CU) $\times n$	0.373[ミリ秒] $\times n$

n : 通信中の端末のペア数

- IPアドレス取得時間
 - DHCPの2往復シーケンス
 - 2重アドレスチェック
- 移動通知時間 (CU)
 - CUの動作がシンプル
 - カーネル内で処理を行っている
- 移動通知時間 (CU) は極めて短い
 - 通信切断時間のほとんどの時間はIPアドレス取得時間

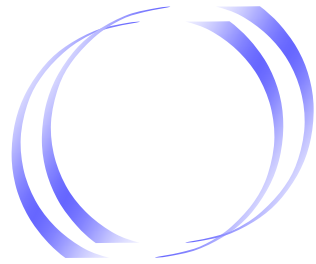


評価② (FTPの性能評価)

- それぞれの状態でPTがCNから100MのファイルをFTPでダウンロードした時間

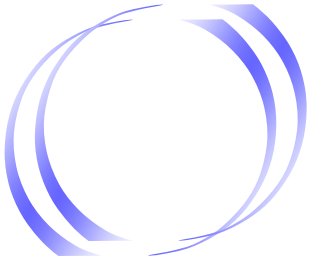
状態	ダウンロード時間	割合
(1)Mobile NPC未実装	129[秒]	1.00
(2)MPRへMobile NPC実装 (アドレス変換なし)	129[秒]	1.00
(3)MPRへMobile NPC実装 (アドレス変換あり)	131[秒]	1.02

- (1)と(2)はほぼ同じであり、(3)は約2%の増加
 - MPRがMobile NPCの機能を保持したことによる中継性能の影響がほとんどない



NAT越え問題

- **移動ネットワークがプライベート空間**
 - 外部から内部の端末にアクセス開始ができない
 - 既存のNAT越え技術と組み合わせ
- NAT-f (NAT-free protocol) と組み合わせることで解決



むすび

- Mobile NPC
 - 概要と動作
 - 実装
 - 評価
- 今後
 - NAT-fとの融合を実装し検証