

モバイル端末の移動透過性を実現する Mobile PPC の実装

竹内 元規† 鈴木 秀和† 渡邊 晃†

† 名城大学大学院理工学研究科

Mobile PPC (Mobile Peer to Peer Communication) は、特別な位置管理サーバを必要とせずにモバイル端末の移動透過性を実現する通信方式である。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でも、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。今回は、本通信方式の試作システムを IP 層に実装し、性能評価をしたので報告する。

Implementation of Mobile PPC realizing the mobility of mobile terminals

Motoki Takeuchi † Hidekazu Suzuki † AKIRA WATANABE †

† Graduate School of Science and Technology, Meijo University

We have proposed the new communication system called Mobile PPC (Mobile Peer to Peer Communication), which can keep their connections during their communications even though they change their locations, without using any extra devices. We have implemented Mobile PPC in IP layer, and evaluated the system.

1 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。また、ホットスポット等の無線ネットワーク環境が整備されつつある。このような状況から、通信中に移動を行っても、通信に影響を与えない方式が要求されている。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。

そこで、様々な工夫により、移動透過性を実現する方式が検討されている[1]。

移動透過性を大きく分類するとプロキシ方式とエンドツーエンド方式がある。プロキシ方式は、通信相手からのパケットをプロキシサーバ

が中継し、移動ノード宛に転送を行う手法で、Mobile IP[2-6]などが提案されている。

エンドツーエンド方式はプロキシを用いずエンド端末間による移動透過な通信を行う方式で、An End-to-End Approach to Host Mobility[8], LIN6[9], MAT[10]などがある。

プロキシ方式とエンドツーエンド方式の両者の特徴を持つものに Mobile IPv6[7]がある。

Mobile IP は、プロキシとして移動ノードの位置を管理するホームエージェント(以下 HA)を導入する。移動ノード宛のパケットを HA が受信し、移動ノードの移動先に届くように、トンネリング転送を行う。移動ノードから通信相手ノードへのパケットは直接送信する。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ーヘッド、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、移動ノードが新しく取得した IP アドレスを直接通信相

手へ通知することができるため、経路の冗長、オーバーヘッドなどの課題は緩和している。しかし、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らない。

An End-to-End Approach to Host Mobility は、エンドツーエンド方式をトランスポート層におけるアプローチで解決する方式である。これは、TCP のオプションを導入し、移動ノードの IP アドレスが変化した際には、TCP オプションによって通知を行い、エンド端末で TCP コネクションを張り直すことで通信を継続させる。この方式では、TCP の拡張が必要であり、またアプリケーションは TCP に限定される。

エンドツーエンド方式をネットワーク層におけるアプローチで解決しているものとして LIN6, MAT がある。これらの方式では、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を設け、ノード識別子と位置指示子の機能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、LIN6 では、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り当てが必要となるという課題がある。また、IPv4 へは適用ができない。MAT は、ホームアドレスとモバイルアドレスという 2 つの IP アドレスを保持させ、前者をノード識別子、後者を位置指示子として両者に対応付ける方式で、通常の IP アドレスを使用することができる。しかし、DNS に独自のレコード追加する必要があり、MAT 非対応のノードは、ホームネットワーク上にいない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ある。LIN6, MAT とも、IP アドレスの対応を保持するために特別な位置管理サーバが必要である。

今後のユビキタス社会を想定するとネットワークの特徴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、プロキシ方式における特殊な装置の存在は、P2P 通信普及の阻害要因となる可能性がある。また、新たなネットワーク機器による基盤が必要となると十分な普及に至るまでその機能が発揮できない。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提供できるこ

とが望まれる。

筆者らは、エンドツーエンド方式をネットワーク層におけるアプローチで解決する方式として、エンド端末の IP 層にアドレス変換処理機能を挿入し、移動前後で IP アドレス変換を行うことで IP アドレスの変化を上位ソフトウェアから隠蔽する通信方式 Mobile PPC (Mobile Peer to Peer Communication) を提案してきた。本稿では、Mobile PPC を FreeBSD 上に実装し、動作確認を実施したので報告する。

以下、2 章で Mobile PPC の動作概要、3 章で Mobile PPC の実装、4 章で試作システムによる移動透過な通信の動作確認、5 章にむすびについて述べる。

2 Mobile PPC

2.1 位置づけ

IP アドレスの変化にかかわらず、通信を可能にするためには、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 アドレスの解決と呼ぶ)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 アドレスの解決と呼ぶ)の 2 つを解決する必要がある。

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)[11-12]という技術が既に実用になっている。本提案システムでは、初期 IP アドレスの解決には DDNS を使用する。

継続 IP アドレスを解決する手段として、以下に述べる Mobile PPC (Mobile Peer to Peer Communication)を適用する。

2.2 Mobile PPC の動作

Mobile PPC では、エンド端末の IP 層で移動の通知処理、アドレス変換処理を行う。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) を保持する。コネクション識別子と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、プロトコル番号の 5 つの情報のことを示す。CIT レコードは、通信が行われた際にコネクション単位で生成され、IP アドレス変換処理は、CIT を参照して行われる。エンド端末間の通信が終了し、無通信状態にある通信の CIT レコードは、タイマにより削除される。

図 1 に Mobile PPC による移動情報の通知方法を示す。移動ノード(MN)が通信相手ノード(CN)と通信中に別のネットワークへ移動する

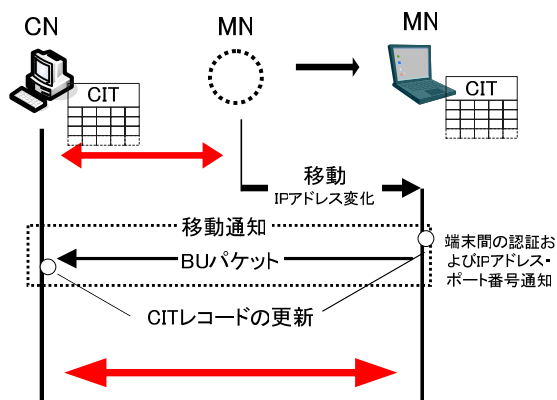


図 1 移動情報の通知

Fig.1 The notice of move information

と、MN は移動先で新しく 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。

この時、一般には通信の乗っ取りを防止するための認証機構について考慮する必要があるが、今回はこの点についての説明は省略する。ここでは、CN と MN 間であらかじめ共有鍵を保持していることを想定し、認証が可能であることとする。

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。

アドレス変換は、移動中に通信が行われていたコネクションに対してのみ行われる。MN が移動後に新しく開始される通信は、移動後に取

得した IP アドレスで行われるためアドレス変換の必要はない。

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。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。

2.3 Mobile PPC の特徴

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、エンドツーエンド方式によるアプローチであり、プロキシのような特殊な装置は不要である。

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変更する必要が無い。

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も生じない。

通常の IP アドレスを使用するので、IPv4 でも IPv6 でも適用可能である。

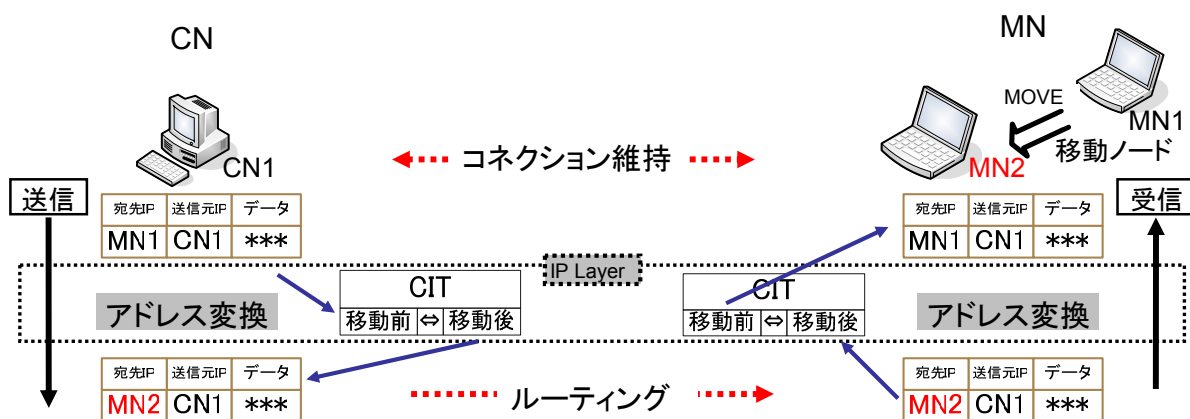


図 2 アドレス変換の例

Fig.2 The example of address translation

3 Mobile PPC の実装

以下に Mobile PPC の実装方式を述べる。

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機能を表 1 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。

Mobile PPC におけるモジュール構成を図 3 に示す。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_output 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

CIT は、通信開始時および移動時の接続識別子 (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 のサイズは 2048 レコードである。

CIT のフォーマットを図 4 に示す。CIT はハッシュテーブルとして実装し、検索キーは通信開始時の接続識別子となる 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ーン法を用いおり、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。

カウンタ値は、CIT 削除デーモンにより定期的にデクリメントされる。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、そのレコードは削除される。レコードが削除される前に通信が発生したら値は初期化される。

3.3 移動の通知

BU のパケットフォーマットを図 5 に示す。BU は ICMP Echo Request をベースに定義されており、MN が移動先における IP アドレス取得処理をトリガーとして生成・送信する。ICMP のデータ部分には、移動情報が付加されている。

表 1 Mobile PPC のモジュール機能
table.1 Function table of Mobile PPC

モジュール	機能
アドレス変換	送信／受信パケット毎に呼び出されるモジュール。 CIT レコードの内容にしたがって、アドレス変換処理やそれにもなうチェックサムの再計算を行う。
移動管理	移動の通知処理を行うモジュール。 自端末の IP アドレス変更時に、BU パケットを生成し通信相手に移動情報を通知する。
CIT 操作	CIT を管理するモジュール。 CIT レコードの検索・生成・更新を行う
CIT 削除デーモン	CIT を監視し、無通信状態のレコードを削除

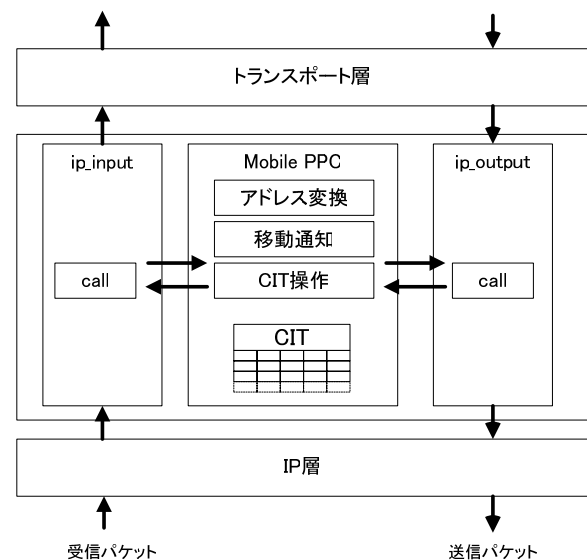


図 3 モジュール構成
Fig.3 Processing in IP layer

検索キー

sIP	dIP	sport	dport	proto	trans	cnt	tsIP	tdIP	tsport	tdport
MN1	CN	XX	YY	TCP	ON	128	MN2	CN	XX	YY
CN	MN2	YY	XX	TCP	ON	128	CN	MN1	YY	XX

図 4 CIT のフォーマット
Fig.4 CIT Format

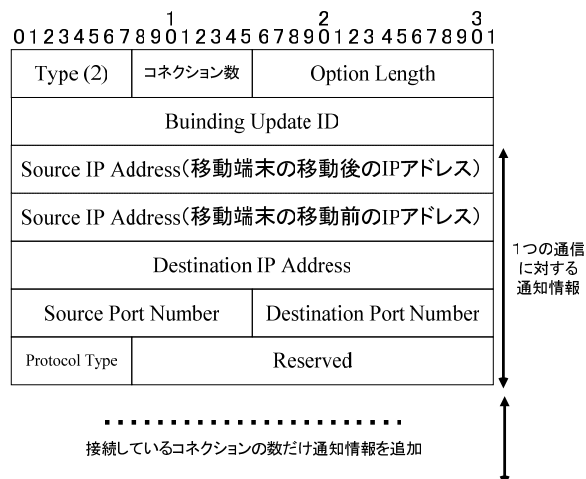


図 5 BU パケットフォーマット
Fig.5 BU packet format

通知する移動情報は、MNが移動後に取得した IP アドレスとエンド端末間で行われていた全通信のコネクション識別子である。BUを受信した CN は、通知された移動情報を元に CIT を更新する。

4 Mobile PPC の動作確認

4.1 移動透過性の確認

移動透過性の確認を図 6 に示す実験環境で行った。Mobile PPC を実装した MN と CN の装置仕様を表 2 に示す。

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但其後もデータ通信が継続していることを確認した。

4.2 処理時間の測定

(1) 通信中断時間

MNがネットワークを移動し、通信を継続するまでの間、通信中断時間が発生した。この時間は、MNがDHCPでIPアドレスを取得する時間とエンド端末間で行われるBUによる通知処理時間の合計である。

表 3 に DHCP による IP アドレス取得時間、表 4 に通知処理時間を示す。通知処理時間には、MN の CIT 更新時間、BU パケットの伝達時間、CN の CIT 更新時間が含まれる。Mobile PPC では、コネクション単位による通信の継続を行うため、エンド端末間で行われているコネクシ

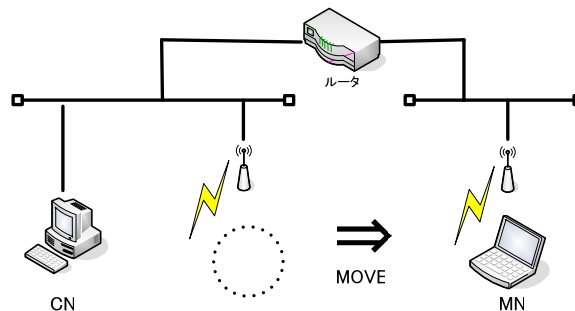


図 6 実験環境
Fig.6 Experimental environment

表 2 装置仕様
table.2 Device specification

	移動ノード	通信相手ノード
CPU	Celeron 2GHz	Pentium プロセッサ 2.4GHz
メモリ	256M	256M
NIC	IEEE802.11b	100BASE-T

表 3 DHCP による IP アドレス取得時間
table.3 IP address acquisition time by DHCP

	DHCP による アドレス取得時間
最大	9.01[秒]
平均	6.33[秒]
最小	4.11[秒]

※10 回試行

表 4 通知処理時間
table.4 Notice management time

エンド端末の コネクション数	1	2	3	4
平均時間[ミ秒]	4.61	4.57	4.72	5.16

※10 回試行の平均

ン数が多ければ通知処理時間も増加する。

通信処理時間がミリ秒単位なのに比べ、DHCP による IP アドレス取得時間は平均で 6.33 秒という時間であるため、通信中断時間の大半は、IP アドレスの取得時間によるもので

ある。

(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5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5より、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。

(3) FTP による性能測定

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からCNへ10M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図7に示す。実装していない状態を基準とすると、Mobile PPC 実装時でアドレス変換なしの状態でも0.5%、アドレス変換をしている状態でも0.8%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 によるオーバーヘッドは実用上、ほとんど影響がないと考えられる。

表5 1パケットごとの処理時間
table.5 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

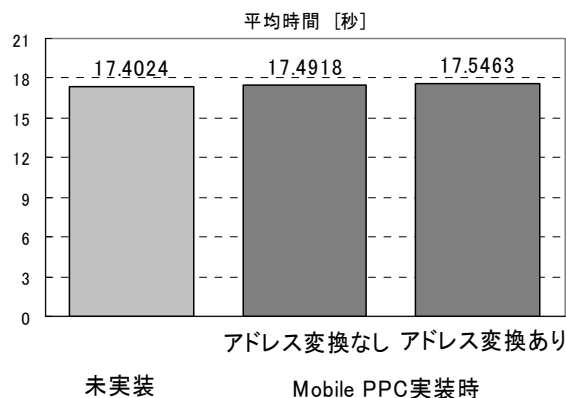


図7 FTP によるダウンロード時間
Fig.7 Download time by FTP

5 むすび

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確認した。

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の3点が挙げられる。移動時には通信の乗っ取りを防止するためにエンド端末にあらかじめ保持している共有鍵を用いた認証おこなっているが、グローバルな環境でも認証ができるような認証機構の定義が求められる。端末がネットワークを移動し、通信が再開されるまで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ー処理が必要となる。移動ノード同士の通信では、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、対策が必要である。

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。

謝辞

本研究は柏森財団の助成を受けて実施したものである。

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，Vol.J87-D-I，No.3，pp.308-328(2004)
- [2]. Perkins,C.：IP Mobility Support for IPv4, RFC3344,IETF,Aug.2002
- [3]. Perkins,C.：IP Encapsulation within IP", RFC 2003, October 1996
- [4]. Calhoun,P. and Perkins,C.：Mobile IP Network AddressIdentifier Extension, RFC 2794, March 2000.
- [5]. C. Perkins, P. Calhoun：Mobile IP Challenge/Response Extensions. RFC 3012.November 2000.

- [6]. G. Montenegro : Reverse Tunneling for Mobile IP, revised RFC3024, Jan. 2001.
- [7]. Johson,D.B. and Perkins,C. : IP Mobility Support in IPv6 , Internet-draft , TETF , Nov.2002
- [8]. 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
- [9]. Ishiyama , M. , Kunishi,M., Uehara,K, Esaki.H,and Teraoka .F , : LINA : A New Approach to Mobiiity Support in Wide Area Networks , IEICE Trans. Commun. , Vol.E84-B , No.8 , PP.2076-2086 (2001)
- [10]. 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.
- [11]. R. Droms, “Dynamic Host Configuration Protocol”, RFC2131, March 1997.
- [12]. Vixie (Ed.), P., Thomson, S., Rekhter, Y. and J. Bound, : "Dynamic Updates in the Domain Name System", RFC 2136, April 1997.

モバイル端末の移動透過性を 実現するMobile PPCの実装

名城大学大学院理工学研究科

竹内 元規

鈴木 秀和

渡邊 晃

はじめに

- モバイル端末の普及
- 無線ネットワーク環境の整備
 - 自由に移動しながら通信したいというニーズ
- 移動を行うと通信の継続が不可能
 - ノードを識別するIPアドレス自体に位置の情報が含まれる
 - ネットワークを移動するとIPアドレスが変化

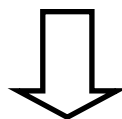


移動ノードのIPアドレス？

・通信開始不可能

移動前後でIPアドレスの組が変化

・上位層で、別の通信と見なされてしまうため通信継続不可能



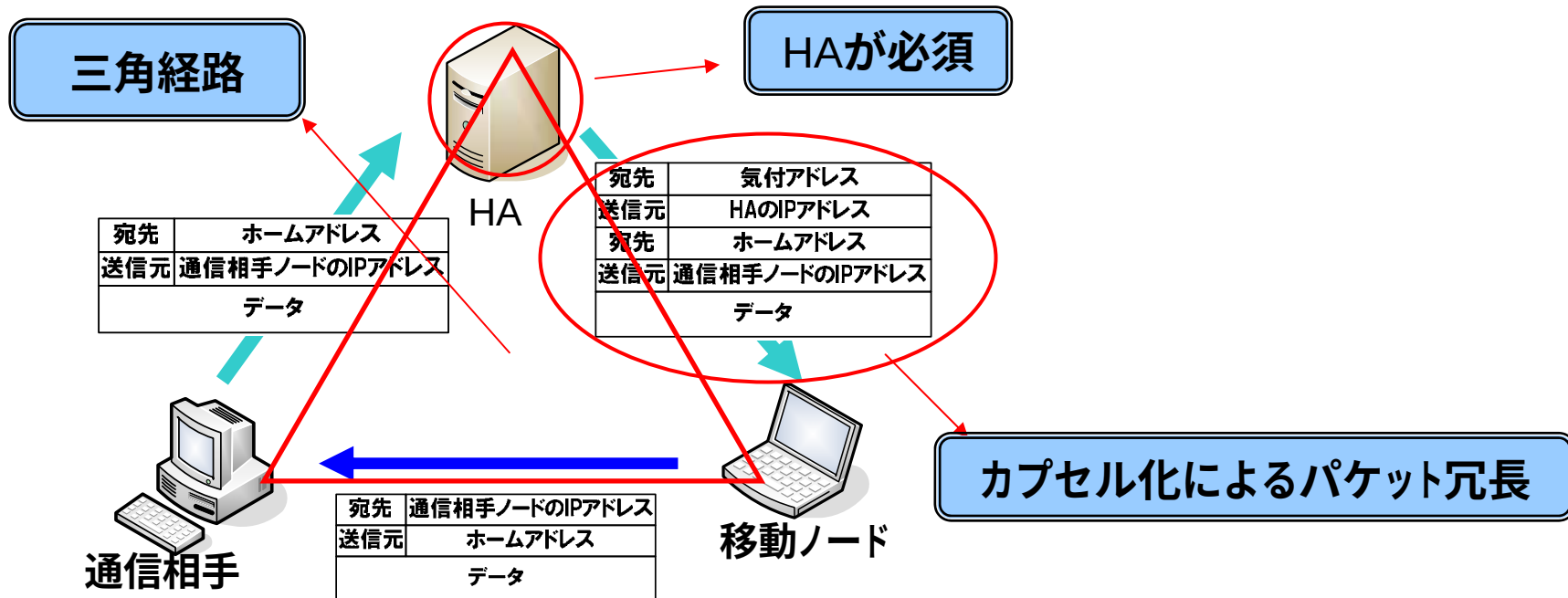
様々な工夫により移動透過性を実現する方式が検討

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



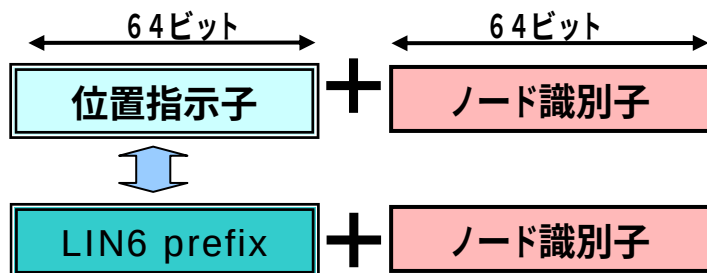
移動透過性を実現する技術とその課題②

➤ 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

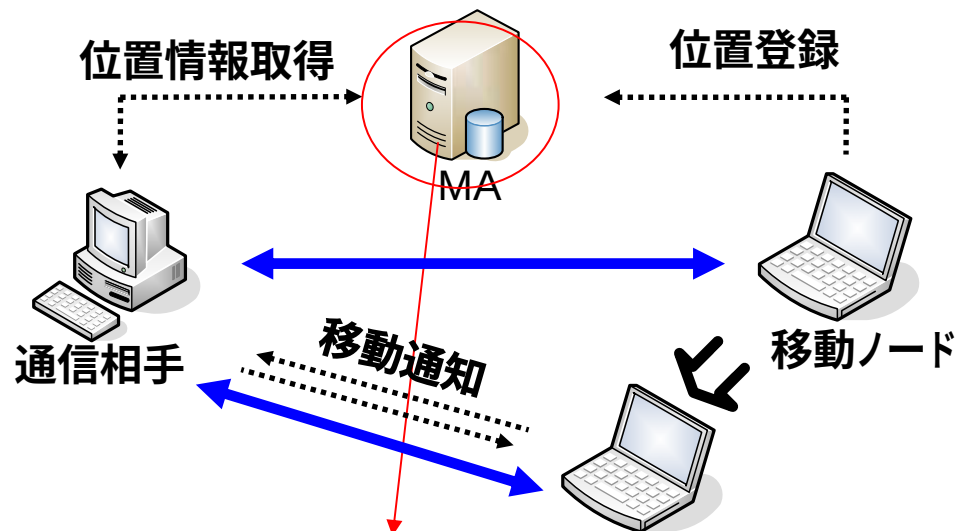
IPv6アドレスを位置指示子とノード識別子に分離した縮退アドレスモデルを導入



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

位置指示子とノード識別子の対応を保持



特殊な位置管理エージェント

提案する通信方式

▶ 今後のユビキタス社会を想定すると

P2P通信の要求が増加

プロキシ方式では、
P2P通信普及の阻害要因になる可能性がある

個人間の通信が主体

エンドツーエンド方式であっても
特殊な位置管理サーバが必要となると、十分
な普及に至るまでその機能を発揮できない

エンドツーエンド方式でかつ、既存の仕組みをできるだけ
変えない方式が求められる

解決

Mobile PPCによる移動透過性を提案している

位置づけ

- **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
 - 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

2つを明確に分離する
- **通信開始時にはダイナミック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
- **移動時における通信の継続にはMobile PPCを適用**

Mobile PPCの概要

Mobile PPC (MOBILE Peer to Peer Communication)

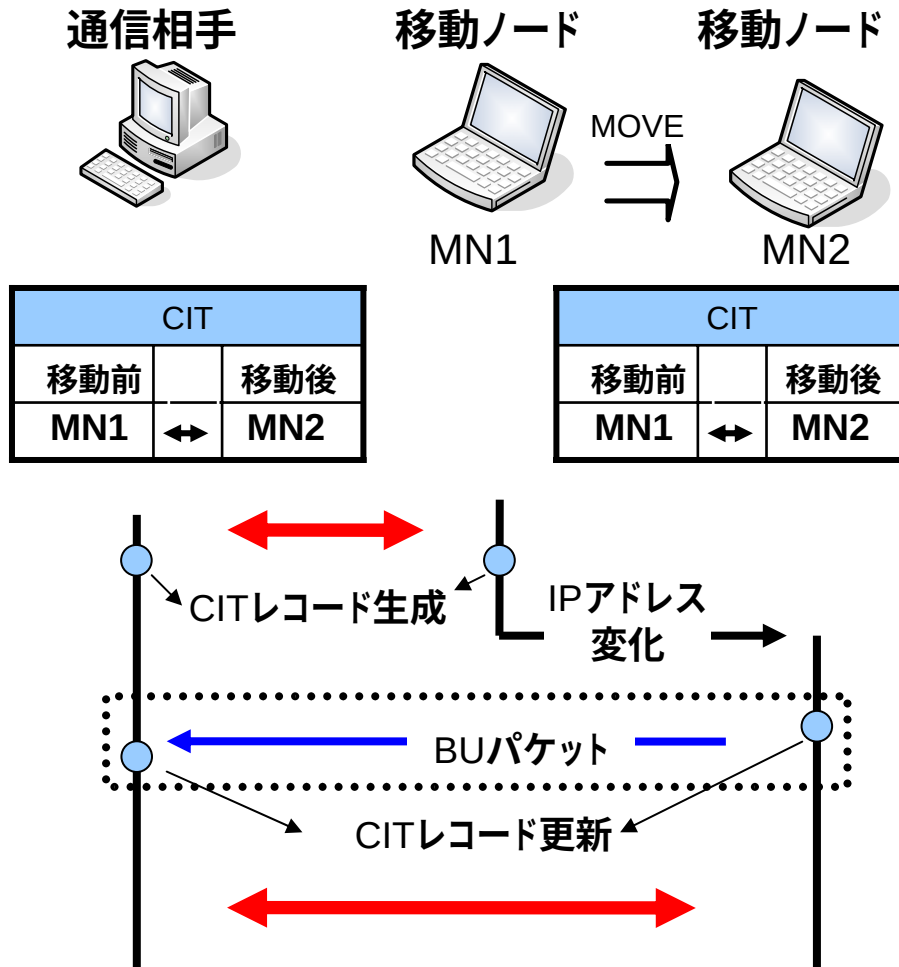
エンドツーエンド方式をネットワーク層におけるアプローチで解決

- **対応ノードは、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**
 - 両端末で行っている通信のIPアドレス, ポート番号, プロトコル番号...

エンド端末のIP層で

- **移動通知処理**
 - 移動時にBU(Binding Update)パケットで移動情報を通知
 - CITテーブル更新
- **IPアドレス変換処理**
 - 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

- 両端末でCITレコード生成

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
- BUパケットの生成・送信

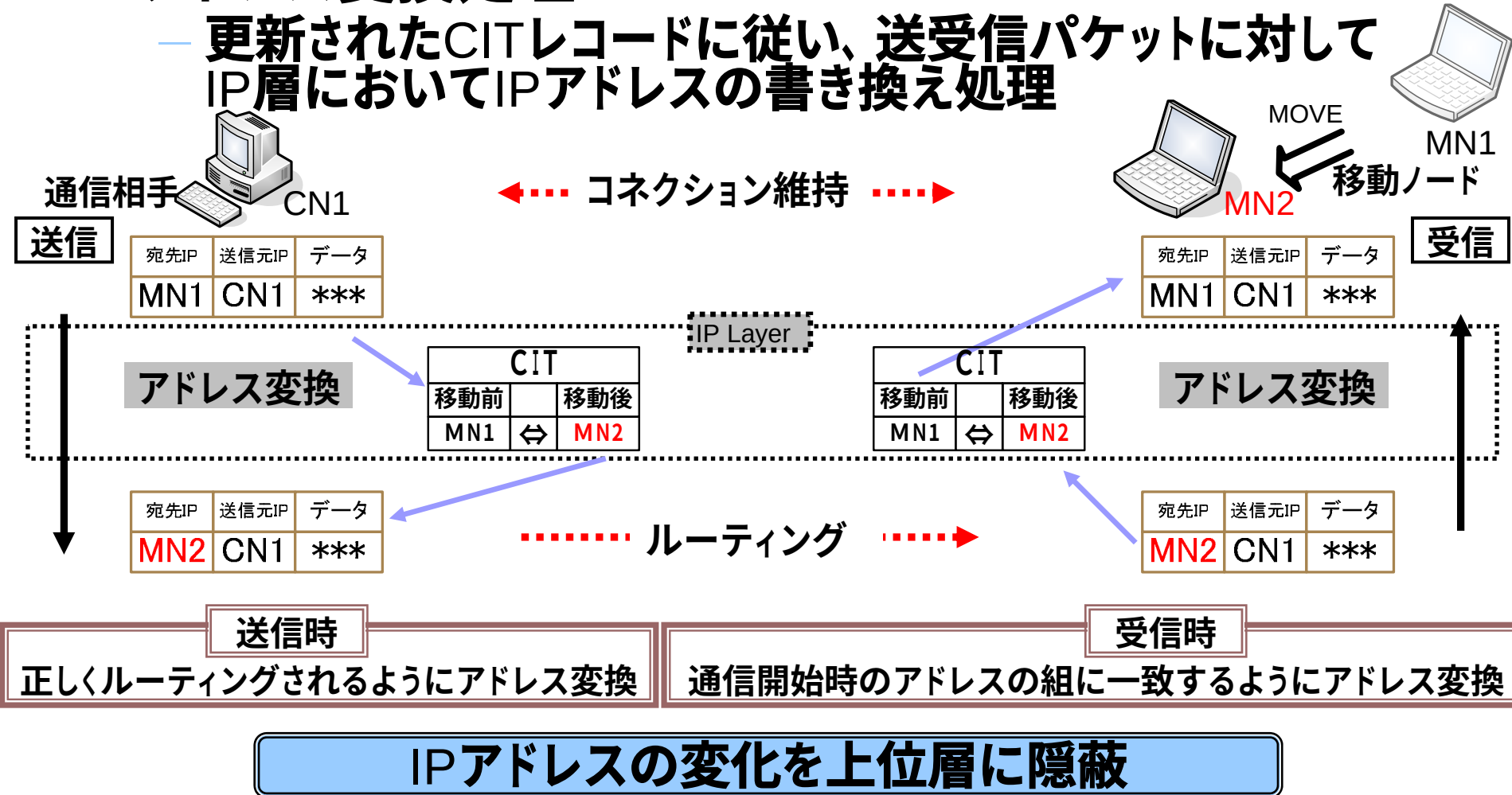
通信相手ノードの処理

- BU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

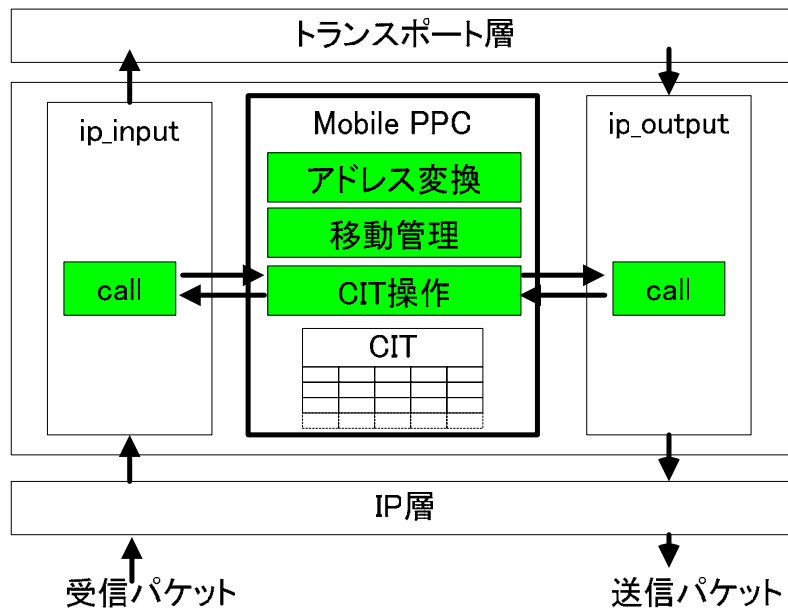
➤ アドレス変換処理

- 更新されたCITレコードに従い、送受信 packets に対して IP 層において IP アドレスの書き換え処理



➤ 実装方式

- FreeBSD(5.2.1-R)のカーネル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



アドレス変換

IPアドレス変換, チェックサム差分計算

移動管理

BU生成・通知

CIT操作

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(チェーン法)

➤ DHCPによるIPアドレス取得に対応

- MNは, DHCP ACK受信時にCITを更新し, BUパケットを生成・送信

➤ BU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

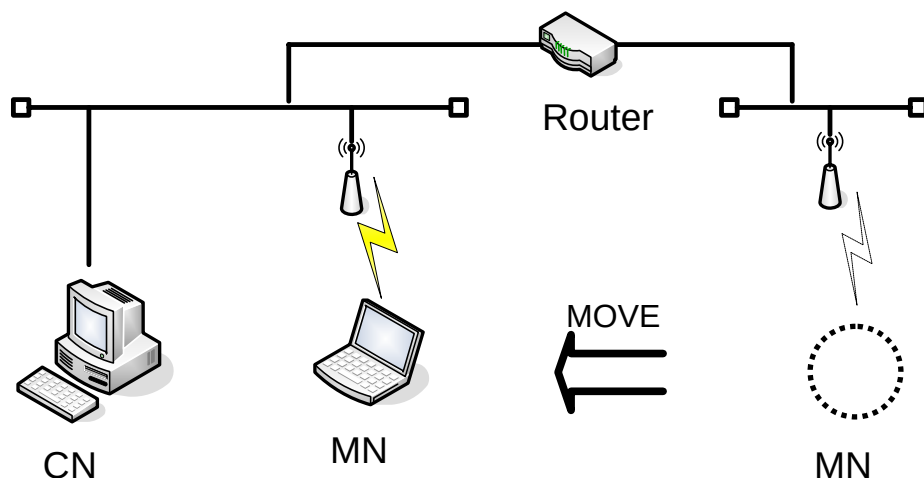
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1																			
Type (2)					コネクション数										Option Length				
Buinding Update ID																			
移動端末の移動後のIPアドレス																			
移動端末の移動前のIPアドレス																			
Destination IP Address																			
Source Port Number										Destination Port Number									
Protocol Type					Reserved														

コネクション数だけ追加

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



	MN	CN
CPU	Celeron 2GHz	Pentium4 2.4GHz
メモリ	256M	256M
NIC	IEEE802.11b	100BASE-T

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

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

➤ 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、アドレス変換をしていない状態
- ③ Mobile PPCを実装し、アドレス変換をしている状態

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 アドレス変換なし	87.31 [秒]
③ アドレス変換あり	87.40 [秒]

➤ 考察

- 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
- 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 PPCの評価

- **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

➤ Mobile PPCにおける課題

1. 移動時の認証機能

- グローバルな環境で利用できる認証機能の検討

2. 移動時のパケットロス

- 移動時の通信中断時間内にパケットロスが発生 (TCPリトライ)
- エンドツーエンド方式の共通の課題として、無線レイヤに関わる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ある
- エンドツーエンド方式の課題として、対策を検討していく

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位置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ーヘッドは、実用上問題ない
 - ネットワーク移動時に、主にIPアドレスの取得時間による通信の中断時間が発生

➤ 今後

- 課題の解決を検討
- IPv6への適用を進めていく

おわり

参考資料

CITフォーマット

- 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

検索キー

sIP	dIP	sport	dport	proto	trans	cnt	tsIP	tdIP	tsport	tdport
MN1	CN	XX	YY	TCP	ON	128	MN2	CN	XX	YY
CN	MN2	YY	XX	TCP	ON	128	CN	MN1	YY	XX

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

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

