情報処理学会研究報告
IPSJ SIG Technical Report

Hole Punchingを用いたNAT越えMobile PPCの実装

鈴木秀和[†1,†2] 寺澤圭史[†1] 渡邊晃[†1]

IPv4 ネットワークでは，移動ノードが通信中にグローバルネットワークとプライベートネットワークを跨って移動することが想定される．筆者らは，通信経路上に NAT が存在した場合，NAT 越え手法として知られている Hole punching の原理を移動透過性技術に適用する方法を検討してきた．これにより，異なるアドレス空間を跨る移動透過性を実現することができる．本稿では，エンドノードだけで移動透過性を実現する Mobile PPC（Mobile Peer-to-Peer Communication）に上記機能を実装したので報告する．

Implementation of NAT Traversal for Mobile PPC Applying Hole Punching

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

In IPv4 network, a mobile node may move between a global address network and a private address network during communication. We have already proposed a mobility mechanism that uses a principle of hole punching, widely known as a NAT traversal technology, when NAT exists on a communication path. It can realize mobility over different types of address areas. I this paper, we describe the implementation of the above function for Mobile Peer-to-Peer Communication (Mobile PPC) that can realize mobility with only end nodes.

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

1. はじめに

携帯端末や無線ネットワークの普及により，ユーザはいつでも，どこからでもネットワークに接続できるようになった．現在は携帯電話網の利用が圧倒的に多いが，近年では，無線 LAN や WiMAX を搭載した携帯端末も登場しており，今後ますますシームレスに IP ネットワークに接続可能なユーザが増加すると考えられる．しかし，IP ネットワークでは通信中にユーザが移動したり，接続する無線デバイスの切り替えをしたりすると，IP アドレスが変化するため通信が切断されてしまう．この問題を解決するために，移動透過性を実現する様々な方式が提案されている[1)]．

移動透過性を実現する技術の多くは，IPv6 の利用を前提としている．しかし，IPv6 は相互接続という点において IPv4 との互換性がないため，IPv4 と IPv6 が混在する環境が当分続くと想定される．通信相手が IPv4 にしか対応していない機器であれば，IPv4 で通信する必要がある．従って，IPv4 ネットワークにおける移動透過性技術の実現は大きな意義があると考えられる．

IPv4 ネットワークではアドレス枯渇問題のため，全ての移動ノード（以下 MN; Mobile Node）にグローバルアドレスを割り当てることは困難であり，プライベートアドレスを積極的に利用する必要がある．そのため，グローバルアドレス空間のネットワーク（以下グローバルネットワーク）とプライベートアドレス空間のネットワーク（以下プライベートネットワーク）を跨った移動があり得る．このような移動では移動前と移動後の通信経路のどちらかに NAT（Network Address Translator）が介在することになる．その結果，移動通知情報に含まれる IP アドレスと実際の通信で用いられる IP アドレスが一致せず，移動前後の IP アドレスの関係を正しく管理できないという課題がある．

Mobile IP[2)] においては，この課題を解決するため，移動通知を UDP によりカプセル化処理を行ったり，NAT に独自の機能を追加するなどの対策が検討されている[3)–5)]．しかし，この方法ではカプセル化処理に起因するヘッダオーバヘッドのため伝送効率が低下したり，特殊な NAT が必要になるなどの課題が残る．

IPv4 ネットワークにおいて，エンドエンドで移動透過性を実現する技術として，Mobile PPC（Mobile Peer-to-Peer Communication）[6)] がある．Mobile PPC では，MN は DDNS（Dynamic DNS）により通信相手ノード（以下 CN; Correspondent Node）の IP アドレスを取得してから通信を開始する．MN が通信中に移動して IP アドレスが変化した際，CN に対して移動前後の IP アドレスの関係を直接通知し，その対応関係を IP 層に記憶する．そ

の後，IP 層で全ての TCP/UDP パケットにアドレス変換処理を行うことにより，上位層から IP アドレスの変化を隠蔽して通信を継続することができる．

筆者らは既存の NAT をそのまま利用でき，かつ MN がアドレス空間の違いに関わらず自由に移動できる方式を提案してきた[7]．提案方式では NAT のアドレス変換に対応するために，NAT 越え技術の代表的な手法として知られている Hole punching[8] の原理を導入する．MN は CN に対して Hole punching に当たる Binding 処理を行うことにより，NAT にマッピング情報を生成し，さらに NAT の外側に割り当てられた IP アドレスとポート番号を取得する．その後，MN は取得した情報を CN 当てに送信する移動通知情報に含めることにより，CN 側で適切なアドレス変換処理を実行させることができる．

以下，2 章で Mobile PPC の基本的仕組みと提案方式について説明する．3 章で提案方式の実装について，4 章で動作確認の結果と残された課題について述べ，5 章にてまとめる．

2. 提案方式の仕組み

2.1 移動パターンと記号の定義

図 1 に IPv4 ネットワークにおける MN の移動パターンを示す．なお，本稿では CN はグローバルネットワークに存在し，MN は必ず通信を開始できることを前提とする．MN の移動パターンは，以下の 4 つが考えられる．

Pattern 1: グローバルネットワークからグローバルネットワークへの移動
Pattern 2: グローバルネットワークからプライベートネットワークへの移動
Pattern 3: プライベートネットワークからグローバルネットワークへの移動
Pattern 4: プライベートネットワークから異なるプライベートネットワークへの移動

従来の Mobile PPC は Pattern 1 のみ対応していたが，提案方式は Pattern 2 から Pattern 4 に対応する機能を提供するものである．

本稿で用いる記号を以下のように定義する．

- Gi; グローバル IP アドレス
- Pi; プライベート IP アドレス
- $A : p$; IP アドレス A，ポート番号 p
- $S \rightarrow D$，$D \leftarrow S$; S から D への通信
- $S \leftrightarrow D$; S と D 間の通信
- $S \Leftrightarrow D$; S から D，または D から S へのアドレス変換

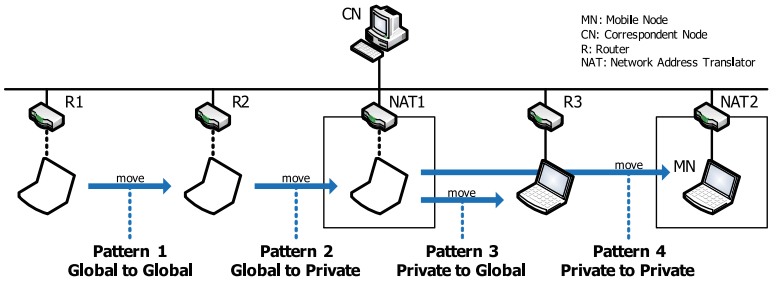


図 1 IPv4 環境における移動パターン
Fig. 1 Mobility patterns in IPv4 environment.

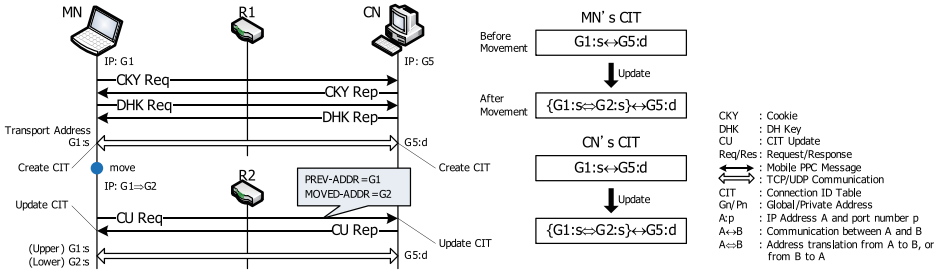


図 2 Mobile PPC の基本シーケンスと生成される CIT
Fig. 2 Basic sequence of Mobile PPC and created CITs.

2.2 Mobile PPC

図 2 に Mobile PPC における通信開始から，移動後の通信継続までの一連の流れを示す．MN および CN の IP アドレスを，それぞれ “$G1$”，“$G5$” とする．MN は通信開始に先立ち，CN と認証鍵共有ネゴシエーションを行う[9]．このネゴシエーションは，Cookie 交換と DH（Diffie-Hellman）鍵交換の 2 往復で構成されており，認証鍵の共有を行う．認証鍵は移動後の移動情報通知処理における認証で用いる．

2 往復のネゴシエーションの後，MN は送信する TCP/UDP パケットのコネクション識別子 CID（Connection ID）[⋆1]を用いて，式 (1) に示すアドレス変換テーブル CIT（Con-

⋆1 TCP コネクション，または UDP ストリームを識別するための情報であり，送信元／宛先 IP アドレス，ポート番号とプロトコルタイプの 5 つの値の組からなる．

nection ID Table）[⋆1]を生成し，通信を開始する．この時点ではパケットのアドレス変換は行われない．CN は TCP/UDP パケット受信時に MN と同様の CIT を生成し，セッション情報を記憶する．

$$MN/CN: \quad G1:s \leftrightarrow G5:d \tag{1}$$

MN が通信中に別のネットワークに移動して，新しい IP アドレス “$G2$” を取得すると，CN との間で移動通知処理を実行する．CN に送信する CU（CIT Update） Request には，移動前の IP アドレス（PREV-ADDR）と移動後の IP アドレス（MOVED-ADDR）が記載され，通信開始時に共有しておいた認証鍵による署名が付加される．

CU Request を受信した CN は認証処理を終えた後，MN の IP アドレスが “$G2$” となるように自身の CIT を式 (2) に示すように更新し，MN に CU Response を応答する．

$$MN/CN: \quad \{G1:s \Leftrightarrow G2:s\} \leftrightarrow G5:d \tag{2}$$

MN は CU Response 受信時に CN と同様に CIT を更新し，移動通知処理を完了する．以後，IP 層において更新した CIT に基づいて TCP/UDP パケットに対してアドレス変換が行われる．

MN は上位層から渡されたパケットの送信元 IP アドレスを，移動前の “$G1$” から移動後の “$G2$” へ変換して CN へ送信する．CN では受信したパケットの送信元 IP アドレスを，MN の移動後の “$G2$” から移動前の “$G1$” へ変換して上位層へ渡す．以上の処理により，上位層から IP アドレスの変化を隠蔽し，かつ正しくルーチングが行われ，通信を継続することができる．

2.3 提案方式

MN と CN の通信経路上に NAT が介在すると，MN が認識・通知する自身のプライベート IP アドレスと，CN が認識する MN の IP アドレス（すなわち NAT のグローバル IP アドレス）が一致しない．その結果，通常の Mobile PPC による移動通知では CIT を正しく更新できず，通信を継続することができない．

上記課題を解決するためには，MN は CN が認識する MN 側トランスポートアドレス，すなわち NAT のグローバル IP アドレスとマッピングされたポート番号を知ればよい．そこで，Mobile PPC に Hole punching の原理を導入する．Hole punching は NAT 越え手法として知られており，Mobile PPC では Binding メッセージを新たに定義してこれを実現する．

⋆1 実際には TCP/UDP プロトコル別に生成されるが，本稿ではプロトコルに関する記載を省略する．

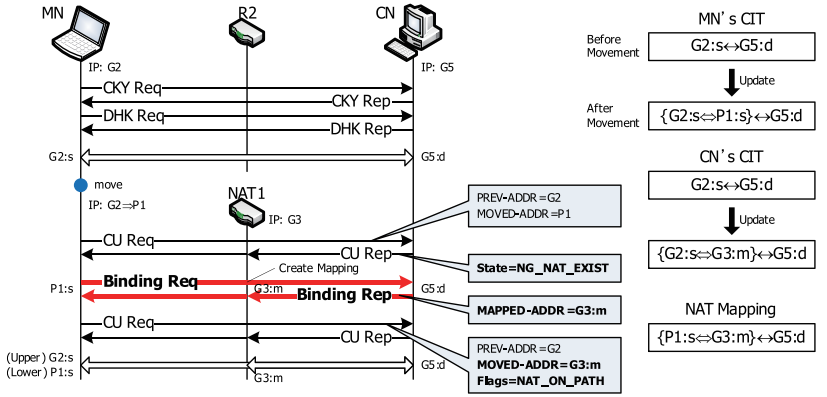


図 3 移動パターン 2 の場合の通信シーケンス
Fig. 3 Communication sequence in the case of mobility pattern 2.

2.3.1 移動パターン 2 の場合

図 3 に MN がグローバルネットワークからプライベートネットワークへ移動する場合の通信シーケンスを示す．通信開始時は通信経路上に NAT が存在しないため，通常の Mobile PPC と同様の処理を行う．

MN がプライベートネットワークへ移動して，新しい IP アドレス “$P1$” を取得すると，CN との間で通常の移動通知処理を実行する．CN に送信する CU Request には，PREV-ADDR “$G1$” と MOVED-ADDR “$P1$” が記載される．CN は CU Request により通知された MOVED-ADDR と，メッセージ送信元 IP アドレス（NAT1 のグローバル IP アドレス “$G3$”）の不一致を検出すると，通信経路上に NAT が存在すると判断して，CIT の更新を行わず，MN に状態フラグ `NG_NAT_EXIST` を設定した CU Response を応答する．

MN は上記フラグが設定された CU Response を受信したら，CN に対して Binding Request を送信する．Binding Request は，移動後の MN が実際に送信する TCP/UDP パケットと同じ IP ヘッダとトランスポートヘッダ，および Mobile PPC ヘッダから構成される[⋆2]．図 3 の場合，送信元が “$P1:s$”，宛先が “$G5:d$” となる．

NAT が Binding Request を転送する際，通常の NAT の原理により送信元トランスポー

⋆2 TCP 通信をしていた場合は TCP ヘッダ，UDP 通信をしていた場合は UDP ヘッダとなる．Mobile PPC ヘッダは Binding メッセージか否かを判断するために記載される．詳細は 3.2 節を参照のこと．

トアドレスを “$P1:s$” から “$G3:m$” に変換し，式 (3) に示すようにマッピング情報を生成する．

$$NAT1: \quad \{P1:s \Leftrightarrow G3:m\} \leftrightarrow G5:d \tag{3}$$

CN は受信した Binding Request の送信元トランスポートアドレス “$G3:m$” を取得し，Binding Response のメッセージ部に MAPPED-ADDR として記載して応答する．MN は取得した MAPPED-ADDR を MOVED-ADDR として，再び移動通知処理を実行する．2 回目の CU Request/Response には，フラグ `NAT_ON_PATH` が設定される．これにより，MN と CN は NAT のアドレス変換を考慮して，正しく CIT を更新することができる．

$$MN: \quad \{G1:s \Leftrightarrow P1:s\} \leftrightarrow G5:d \tag{4}$$

$$CN: \quad \{G1:s \Leftrightarrow G3:m\} \leftrightarrow G5:d \tag{5}$$

2.3.2 移動パターン 3 の場合

図 4 に MN がプライベートネットワークからグローバルネットワークへ移動する場合の通信シーケンスを示す．この場合，通信開始時の通信経路上に NAT が存在するため，移動パターン 2 の移動通知処理と同様の仕組みにより，認証鍵共有処理の途中で Binding 処理を行う．

NAT1 では Binding Request を転送する際，式 (3) と同じマッピング情報が生成される．これにより，MN は予め CN が認識する MN 側トランスポートアドレス “$G3:m$”，すなわち MAPPED-ADDR を知ることができる．

MN がグローバルネットワークへ移動して，新しい IP アドレス “$G4$” を取得すると，CN との間で移動通知処理を実行する．ここで，CU Request に記載する PREV-ADDR は MAPPED-ADDR “$G3:m$” とし，またフラグ `NAT_OFF_PATH` を設定する．これにより，MN と CN は NAT のアドレス変換がなくなったことを考慮して，正しく CIT を更新することができる．

$$MN: \quad \{P1:s \Leftrightarrow G4:s\} \leftrightarrow G5:d \tag{6}$$

$$CN: \quad \{G3:m \Leftrightarrow G4:s\} \leftrightarrow G5:d \tag{7}$$

2.3.3 移動パターン 4 の場合

プライベートネットワークから異なるプライベートネットワークに移動する場合は，移動前と移動後の両通信経路上に NAT が存在することになる．この場合，図 4 の移動前処理と，図 3 の移動後処理を組み合わせる．従って，移動前と移動後に行う Binding 処理により取得した 2 つの MAPPED-ADDR を，それぞれ PREV-ADDR と MAPPED-ADDR に設定することにより対応することができる．

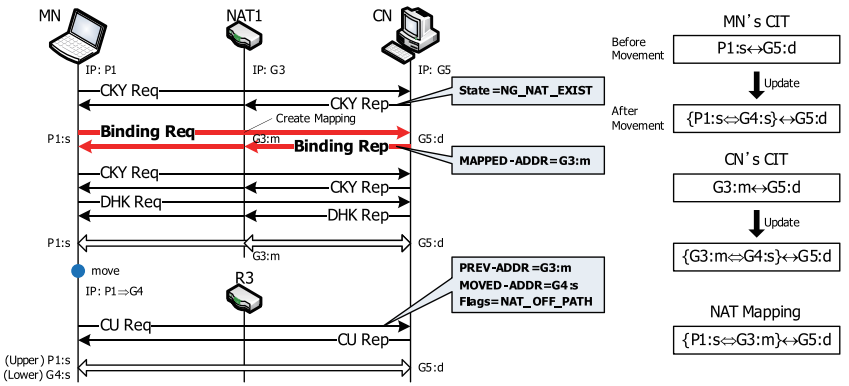


図 4 移動パターン 3 の場合の通信シーケンス
Fig. 4 Communication sequence in the case of mobility pattern 3.

3. 実　　装

3.1 モジュール構成と処理フロー

図 5 と**図 6** に Mobile PPC のモジュール構成と，移動前後のシーケンスとの対応関係を示す．Mobile PPC は IP 層に実装されるカーネルモジュールと，ユーザランドで動作するデーモン（`mppcd`）から構成される．

カーネルモジュールは，CIT 制御部，認証鍵管理部，移動管理部，および今回新たに実装した Binding 制御部がある．Binding 制御部は Binding メッセージの生成および送受信処理を担い，取得した MAPPED-ADDR を認証鍵管理部および移動管理部へ渡す．

アプリケーションが通信を開始すると，TCP/UDP パケットがまず認証鍵管理部へ渡される．ここで，認証鍵を管理するテーブル NIT（Node Information Table）を確認し，対応する通信相手の認証鍵を検索する．認証鍵がない場合は，処理中の TCP/UDP パケットを待避してから認証鍵共有ネゴシエーションを開始する．ここで，受信した Cookie Response に状態フラグ `NG_NAT_EXIST` が設定されていたら，Binding 制御部の処理を割り込ませる．

認証鍵共有ネゴシエーションの 1 往復目を終えると，待避していた TCP/UDP パケットを戻して CIT を生成後，通信を開始する．その後，認証鍵管理部は MPPC ソケット[⋆1]を

⋆1 ユーザランドとカーネルモジュール間のデータ受渡しを実現するために実装したソケットインタフェース．

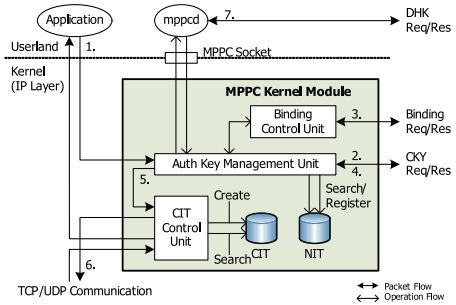


図 5 モジュールと通信シーケンスの関係（通信開始持）
Fig. 5 Diagram relating module to sequence (When MN starts communication).

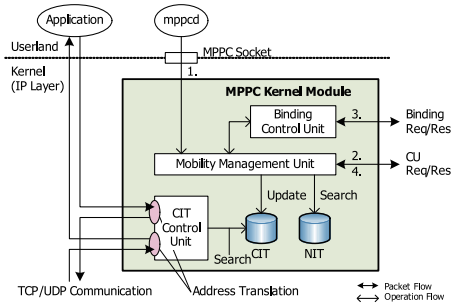


図 6 モジュールと通信シーケンスの関係（移動後）
Fig. 6 Diagram relating module to sequence (After MN moves).

通じて，デーモン `mppcd` に 2 往復目の DH 鍵交換を実行させる．この仕組みにより，DH 鍵交換および認証鍵生成処理を実際の TCP/UDP 通信のバックエンドで動作させることができ，通信開始時のオーバヘッドを削減することができる．

また `mppcd` は移動を検知すると，移動管理部に移動通知処理を行うよう指示する．移動管理部は認証鍵管理部と同様に，受信した CU Response に状態フラグ `NG_NAT_EXIST` が設定されていたら，Binding 制御部の処理を割り込ませる．CIT の更新後，TCP/UDP パケットは CIT 制御部においてアドレス・ポート変換されて，上位層または下位層へ渡される．

3.2 メッセージフォーマット

図 7 に Mobile PPC メッセージフォーマットを示す．Mobile PPC メッセージは ICMP Echo/Echo Reply をベースとしており，ICMP ヘッダ以降に Mobile PPC ヘッダ（図 7(a)）が続く構造になっている．

本提案方式では新たに Node ID と呼ぶ 16 Octets のフィールドを追加する．Node ID とは各ノードを一意に特定するための ID であり，複雑な IP アドレスの変化が発生しても，どのノードからのメッセージであるかを容易に識別することができる．この値は UUID (Universally Unique Identifier)[10] を利用し，ノードの FQDN からハッシュ関数により生成する．

Mobile PPC は DDNS（Dynamic DNS）[11] を利用して通信相手の初期 IP アドレスを解決する．従って，Mobile PPC を利用する全ノードは必ず FQDN を有しており，Node ID の生成に当たり，新たな設定をノードに施す必要はない．

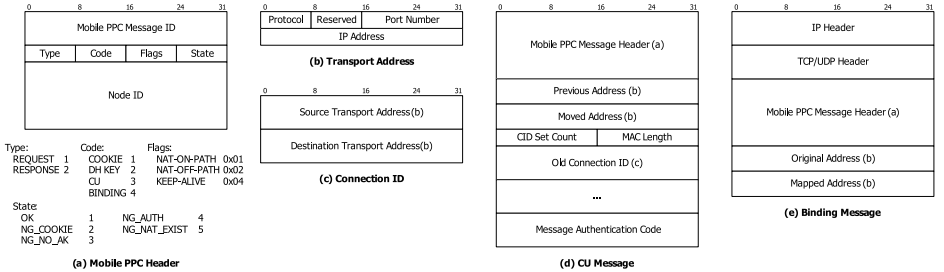


図 7 Mobile PPC メッセージフォーマット
Fig. 7 Mobile PPC message formats.

この他に，CU メッセージにおける PREV-ADDR と MOVED-ADDR の拡張を行った．従来の Mobile PPC では移動前後の IP アドレスだけを通知すればよかったが，提案方式では IP アドレスに加えてポート番号を通知する必要がある．そこで，新たにトランスポートアドレス構造体（図 7(b)）を定義した．これは各ノードが通信に利用するトランスポートアドレス（IP アドレスとポート番号の組）を 1 つのデータセットとし，PREV-ADDR と MOVED-ADDR（図 7(d)）の他，Connection ID（図 7(c)）などで利用する．

Binding メッセージは図 7(e) のように，TCP/UDP ヘッダ以降に Mobile PPC ヘッダが続く．Original Address フィールドには Binding Request の送信元トランスポートアドレスが，Mapped Address フィールドには NAT においてマッピングされた MAPPED-ADDR が記載される．

3.3 Binding Message List の定義

Binding 処理の導入に当たり，新たに BML（Binding Message List）を定義する．**図 8** に BML の構造を示す．BML は Binding メッセージの情報の格納，およびトリガとなったメッセージの待避を主な目的としている．また，BML の作成と同時に Binding メッセージの監視を開始し，BML の削除と同時に監視を終了する．BML は連結リスト構造とし，通信相手毎に BML が生成され，さらに確立しているセッション数分のサブ連結リスト構造をとる．

例えば MN が CN と 2 つのセッションを確立していた場合，移動時に送信する CU Request は 1 つでよく，そのメッセージに 2 セッション分の情報を格納している（図 7(d) 参照）．このように CU は送信すべき情報を集約しているが，Binding 処理は各セッションごとに実行するため集約することができない．

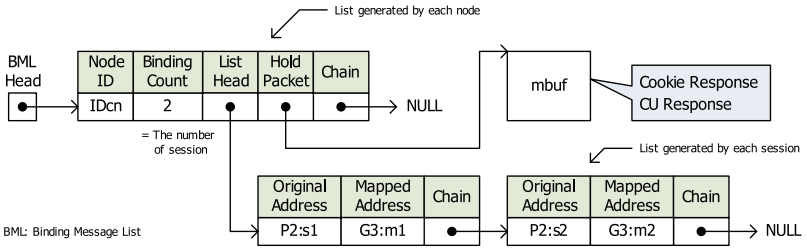


図 8 BML 構造
Fig. 8 Structure of Binding Message List.

そこで，各ノードは CU に集約されているセッションの情報の数を Binding Count として設定し，通信相手から Binding メッセージを受信する度に値をデクリメントしていく．この値が 0 になったら，CN では BML を削除して Binding メッセージの監視を終了する．MN では受信した Binding Response に記載されている 2 つのトランスポートアドレスを，BML のサブ連結リストに記録する．Binding Count が 0 でない場合は，他の Binding メッセージを受信するはずであるため，監視を継続する．Binding Count が 0 になると，記録したトランスポートアドレスを認証鍵管理部または移動管理部へ渡してから，BML を削除する．

4. 動作確認結果と残された課題

4.1 動作確認

今回は FreeBSD 7.0-RELEASE に対して，Binding 制御部の実装および移動管理部の改良を行った．上記機能を実装した MN と CN が NAT 外部のネットワークにおいて，UDP による通信を開始した後，MN が NAT 配下のネットワークへ移動する場合の動作確認を行った．この結果，正しく動作していることを確認し，グローバルネットワークからプライベートネットワークへ移動しても通信を継続することができた．

4.2 残された課題

(1) プライベートネットワークからグローバルネットワークへの移動

認証鍵共有ネゴシエーション時に行う Binding 処理により MAPPED-ADDR を取得するが，このトランスポートアドレスを移動通知処理を行うまで保持する必要がある．そこで，移動前に取得した MAPPED-ADDRESS を NIT エントリに保存しておく．MN は移動するまでに複数の CN と複数の MAPPED-ADDR を取得する可能性があるため，NIT に BML のようなサブ連結リスト構造を追加するような拡張を行う必要がある．

(2) TCP による Binding 処理とファイアウォールの影響

NAT 越えを検討する際，ファイアウォールの存在を無視することができない．近年，多くの NAT ルータには SPI（Stateful Packet Inspection）機能が実装されており，Outbound パケットと Inbound パケットの整合性を検査する．正当な手順の TCP/UDP セッションと判断できない場合は，パケットは破棄される．特に TCP 通信では，ヘッダの SYN フラグや ACK フラグのハンドシェイク状態を記憶するため，Binding シーケンスを 3-way handshake となるような 1.5 往復のシーケンスにする必要があると考えられる．

(3) Keep-Alive

Binding 処理によって生成された NAT マッピング情報は，一定時間の無通信状態が経過すると削除されてしまう．TCP セッションの場合は一度コネクションが確立されると長時間マッピングが維持されるが，UDP の場合は比較的短時間しかマッピングが維持されない[⋆1]．そのため，MN は CN に対して UDP 通信をしている場合，Keep-Alive を定期的に行う必要がある．

5. まとめ

本稿では Hole punching の原理を Binding 処理として，その一部機能を Mobile PPC に実装した．その結果，MN はグローバルネットワークからプライベートネットワークへ移動しても通信を継続できることを確認した．

今後は，残された課題に示した実装を完了させ，通信断絶時間に与える影響や Keep-Alive による負荷の評価などを行う．これまでに，通信相手がプライベートネットワークにいる場合においても，外部からの着信を可能とし，かつ移動透過性を実現できる方式を文献 12) にて提案している．この方式は NAT に機能を追加しているため，今回のように MN が不特定なプライベートネットワークへ移動するケースには対応できない．Binding 機能を有効に活用して，外部からの着信を受けることが可能な方式を別途検討する予定である．

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

⋆1 例えば，FreeBSD の NAT の場合，TCP は 24 時間，UDP は 60 秒でマッピングが破棄される．（http://fxr.watson.org/fxr/source/netinet/libalias/alias_db.c?v=FREEBSD70;im=bigexcerpts）CISCO NAT の場合，TCP は 24 時間，UDP は 5 分である．(http://www.cisco.com/en/US/tech/tk648/tk361/technologies_q_and_a_item09186a00800e523b.shtml#qa45)

参 考 文 献

1) 寺岡文男：インターネットにおけるノード移動透過性プロトコル，電子情報通信学会論文誌 (D-I)， Vol.J87-D1, No.3, pp.308–328 (2004).
2) Perkins, C.: IP Mobility Support for IPv4, RFC 3220, IETF (2002).
3) Levkowetz, H. and Vaarala, S.: Mobile IP Traversal of Network Address Translation (NAT) Devices, RFC 3519, IETF (2003).
4) Montenegro, G.: Reverse Tunneling for Mobile IP, revised, RFC 3024, IETF (2001).
5) 井戸上彰，久保 健，横田英俊：プライベートアドレスを使用するモバイルネットワーク間のローミング手順とその実装，情報処理学会論文誌， Vol.44, No.12, pp.2958–2967 (2003).
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 の提案と実装，情報処理学会論文誌， Vol.47, No.12, pp.3244–3257 (2006).
7) 鈴木秀和，渡邊 晃：Hole Punching を用いた NAT 越え Mobile PPC の設計，情報処理学会研究報告，2008-MBL-45, Vol.2008, No.44, pp.69–74 (2008).
8) Ford, B., Srisuresh, P. and Kegel, D.: Peer-to-Peer Communication Across Network Address Translators, *Proc. USENIX Annual Technical Conference*, Anaheim, CA, pp.179–192 (2005).
9) 瀬下正樹，渡邊 晃：Mobile PPC における認証方式の実装，DICOMO2006 シンポジウム論文集，Vol.2006, No.6, pp.809–812 (2006).
10) Leach, P., Mealling, M. and Salz, R.: A Universally Unique IDentifier (UUID) URN Namespace, RFC 4122, IETF (2005).
11) Vixie, P., Thomson, S., Rekhter, Y. and Bound, J.: Dynamic Updates in the Domain Name System (DNS UPDATE), RFC 2136, IETF (1997).
12) 鈴木秀和，渡邊 晃：プライベートネットワーク内のノードを通信相手とした移動透過性の実現方式，電子情報通信学会論文誌 (B)， Vol.J92-B, No.1, pp.109–121 (2009).

Watanabe Lab.
Department of Information Engineering, Meijo University
Information Networks Laboratory

Hole Punchingを用いた NAT越えMobile PPCの実装

鈴木　秀和†‡　　**寺澤　圭史**†　　**渡邊　晃**†

†名城大学大学院理工学研究科

‡日本学術振興会特別研究員PD

Agenda

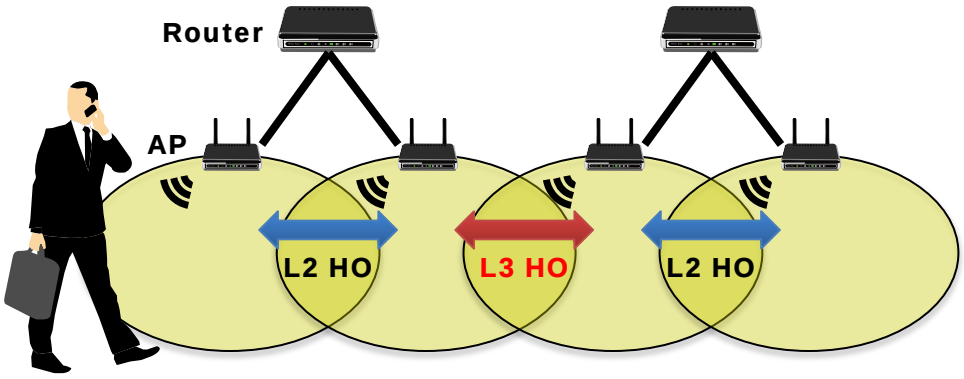
1. **IPv4ネットワークにおける移動透過性**
   - 既存のNAT Traversal手法（Mobile IP）
2. **提案方式**
   - Mobile PPC
3. **実装と動作検証**
4. **今後の課題**

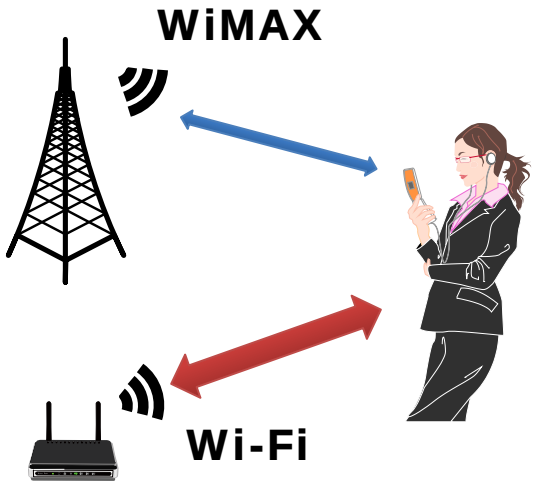
研究背景

- **移動透過性**
  - ノードのIPアドレスの変化をアプリケーションやユーザから隠蔽すること

  ➔IPアドレスが変化しても通信継続が可能

- **IPアドレスの変化が発生する状況**
  1. 基地局の切り替え（ルータをまたぐ場合）
  2. 無線デバイスの切り替え





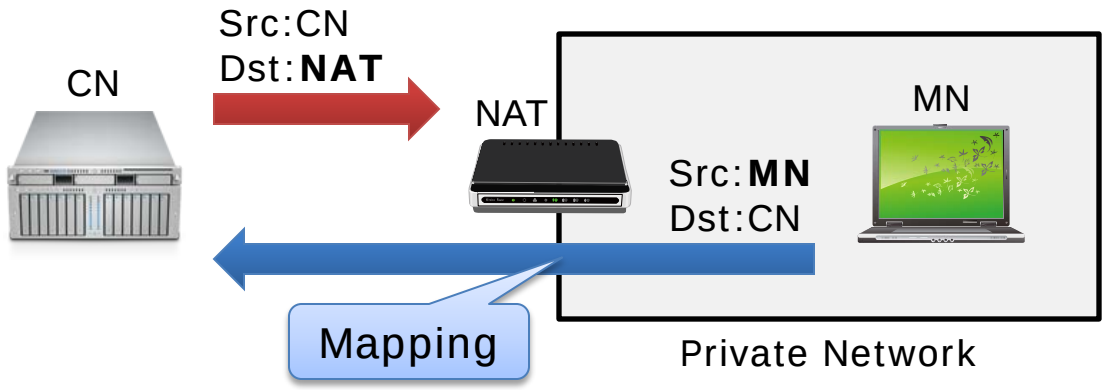
なぜIPv4がターゲットなのか？

- **まず…**
  - IPv6が早く普及してくれたほうがHappy☺
- **しかし…**
  - **接続先ネットワークがIPv6とは限らない**
    - IPv4しか対応していないネットワークはたくさんある
  - **IPv6はIPv4との相互接続性がない**
    - 相手がIPv6非対応ノード➔IPv4通信

“ユビキタス”なモビリティを実現するためには
IPv4環境における移動透過性技術の実現も重要

IPv4環境におけるモビリティ問題

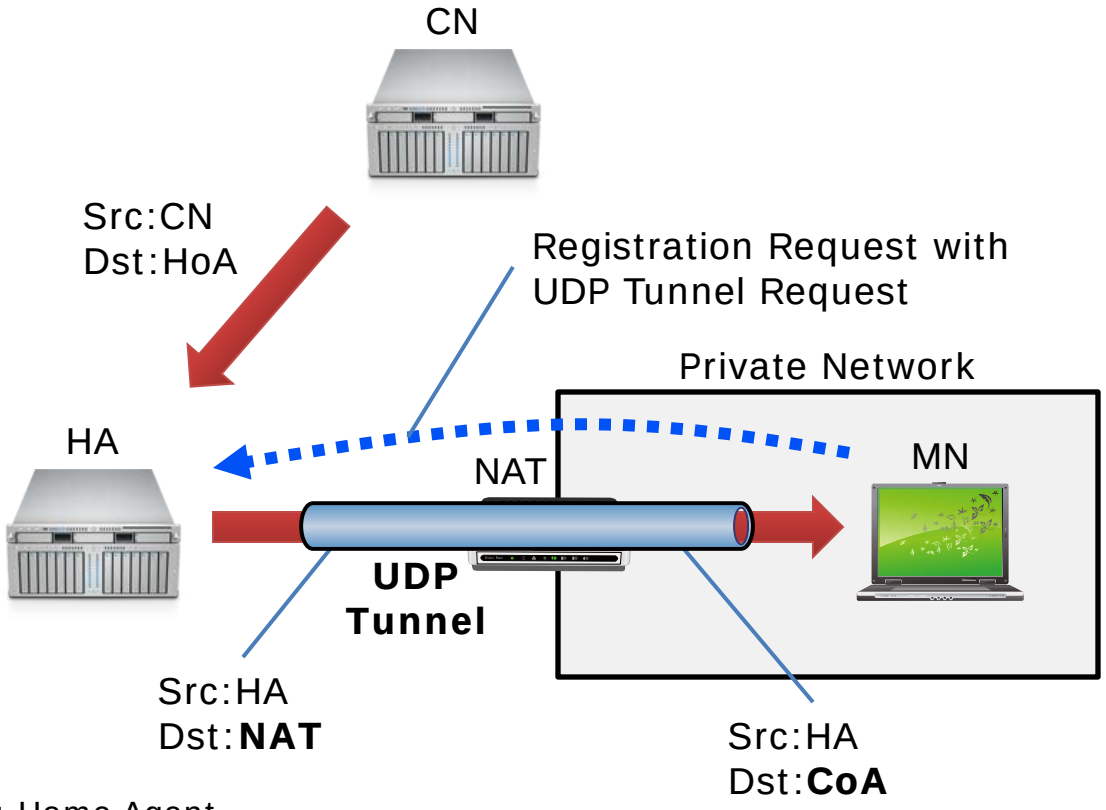
- **MNに割り当てるグローバルアドレスが不足**
  - NATを利用してプライベートIPアドレスを利用
- **コネクティビティの維持が困難**
  - CNはMNに対して直接のリーチャビリティがない
  - NATによるアドレス変換により，コネクション識別情報が非対称



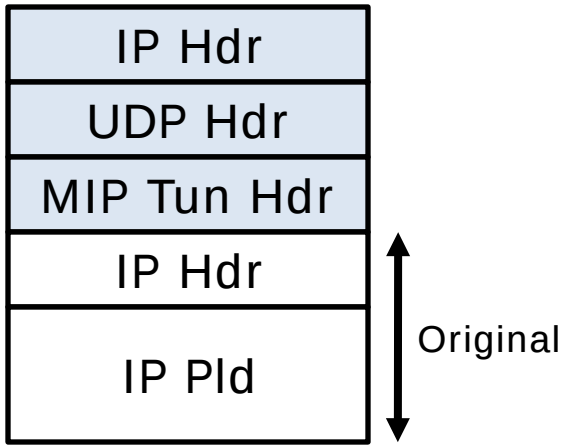
MN : Mobile Node
CN : Correspondent Node
NAT : Network Address Translator

Mobile IPにおけるNAT対策

- **UDPトンネルによるNAT越え**（RFC3519）
  - MNがHAとUDPトンネルを構築し，トラヒックを誘導



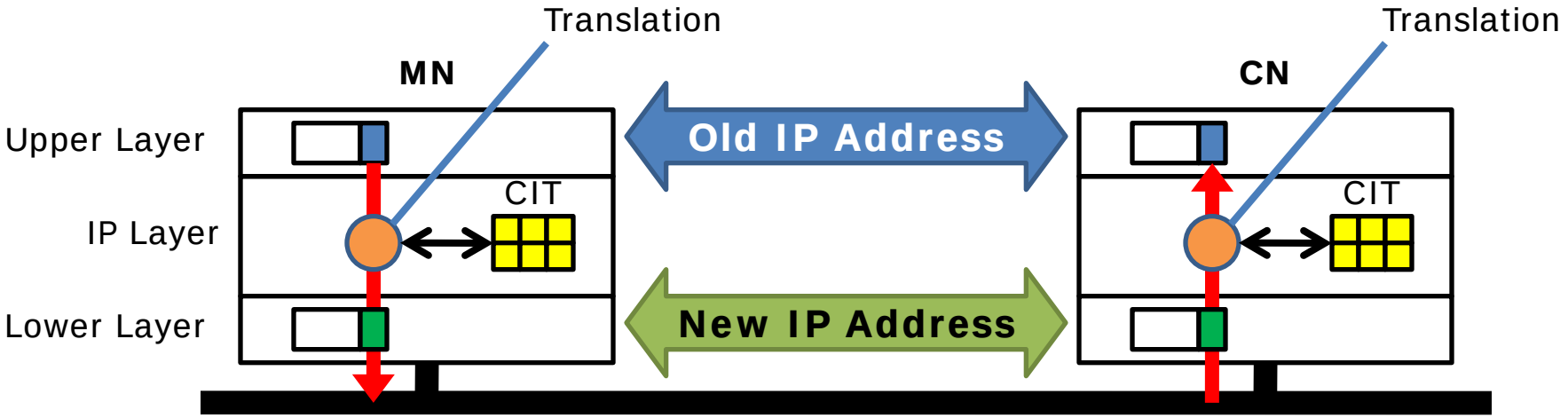
カプセル化処理に起因するオーバヘッドのため，伝送効率が低下



HA ：Home Agent
HoA：Home Address
CoA：Care-of Address

Mobile PPC (Mobile Peer-to-Peer Communication)

- **MNは直接CNに移動通知**
- **両ノードはIP層にアドレス変換テーブルを生成**
  - CIT (Connection ID Table)
- **アドレス変換処理により，アドレスの変化を隠蔽**
  - 送信時：Old IPアドレス➔New IPアドレス
  - 受信時：New IPアドレス➔Old IPアドレス

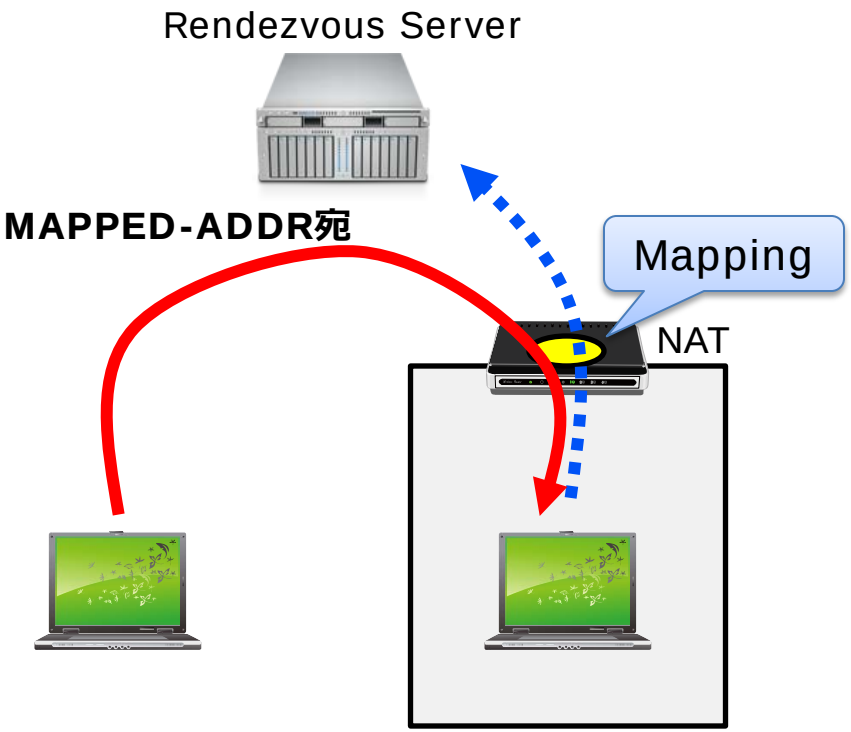


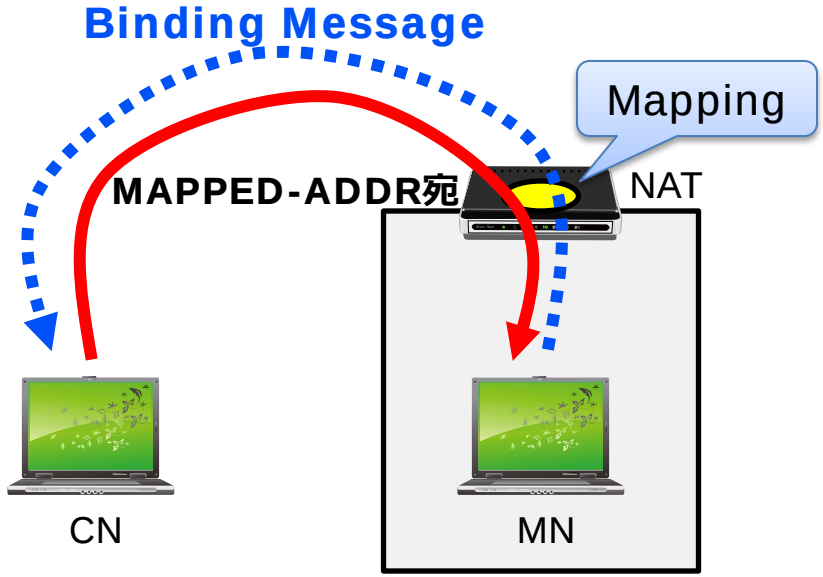
提案方式

- **Hole Punchingの原理を利用**
  - NAT Traversal技術
  - NATの内側から外側にパケットを投げて **“穴”** を開ける

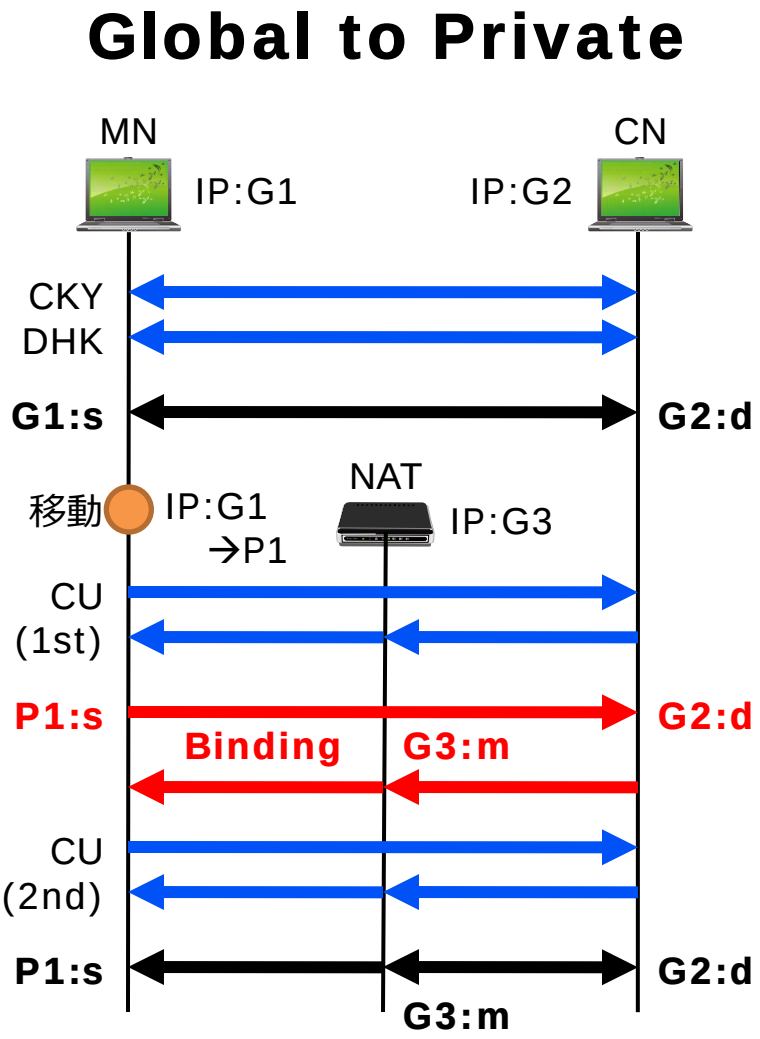
=

MAPPED-ADDR
（NATのIP+ポート番号）





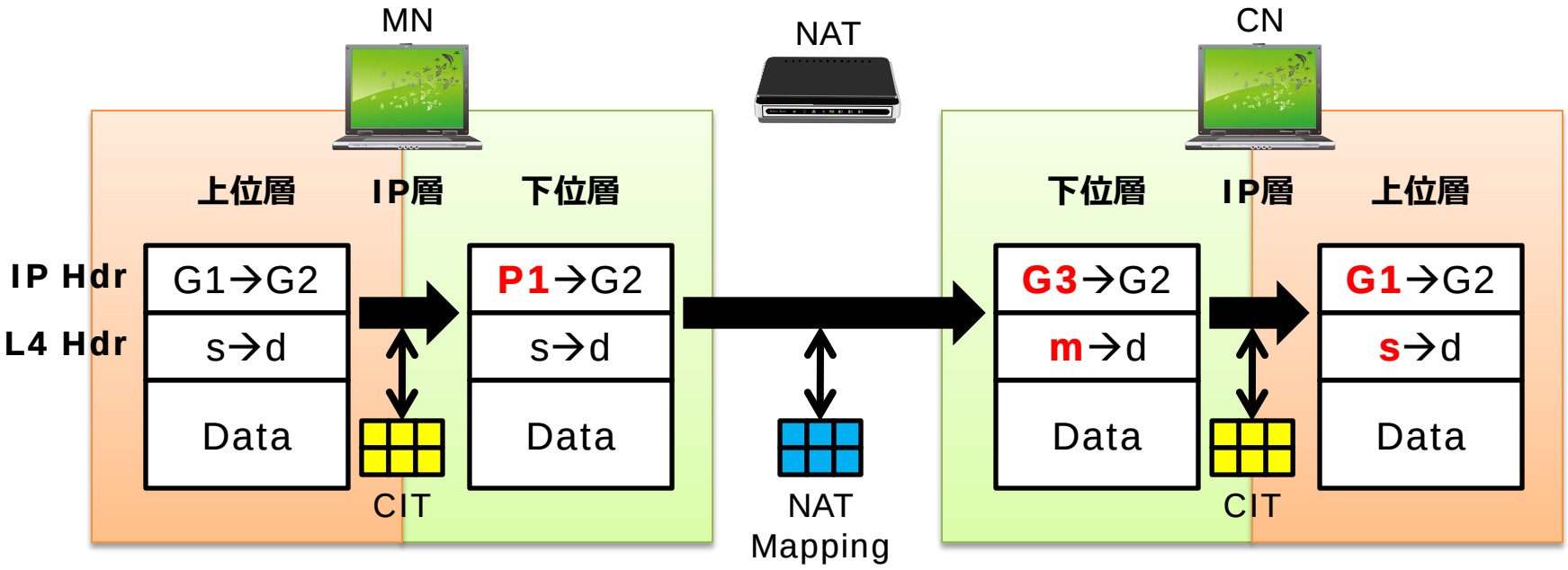
Bindingメッセージの割り込み



- **1回目のCU（CIT Update）**
  - 従来通り
    - 移動前＝G1，移動後＝**P1**
  - CNは**アドレスの不一致を検出**
    ➔Binding処理をMNに要求
- **Binding Req./Res.**
  - MAPPED-ADDR＝G3：m
- **2回目のCU**
  - Binding結果を反映
    - 移動前＝G1：s，移動後＝**G3：m**

CKY: Cookie　DHK: DH Key

パケットのアドレス変化の遷移



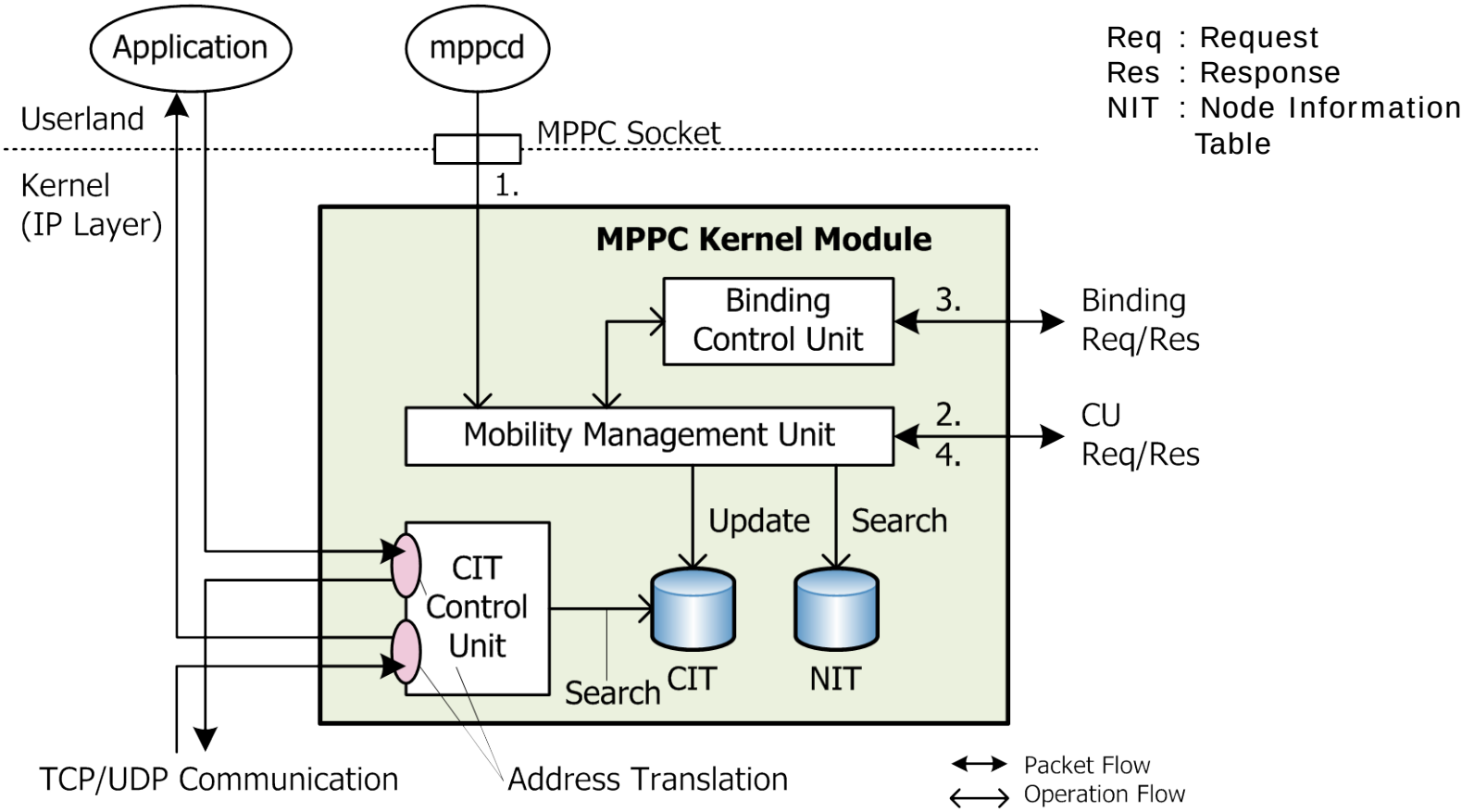
- **CNからMNへの応答**
  - 上記と逆（宛先トランスポートアドレス）の変換を実行

➢ **MNの移動に伴う変化**
➢ **NATの処理に伴う変化**

→ **上位層からいずれも隠蔽しており，通信の継続が可能**

実装

- FreeBSD 7.0-RELEASEのカーネルに実装
  - Binding制御部を従来のカーネルモジュールに追加



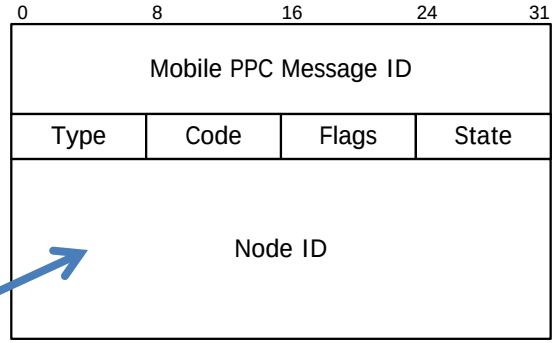
制御メッセージフォーマットの拡張

- **ノードIDの追加**
  - アドレスの変化に影響されずノードを一意に識別
  - **UUID（Universally Unique Identifier）**を利用
    - ノードのFQDNからハッシュ関数により生成

FQDN: alice.example.com

Node ID:
550e8400-e29b-41d4-a716-446655440000

16バイトの数値．重複や偶然の一致が
起こりえないと確認して用いることが可能．
（RFC4122）



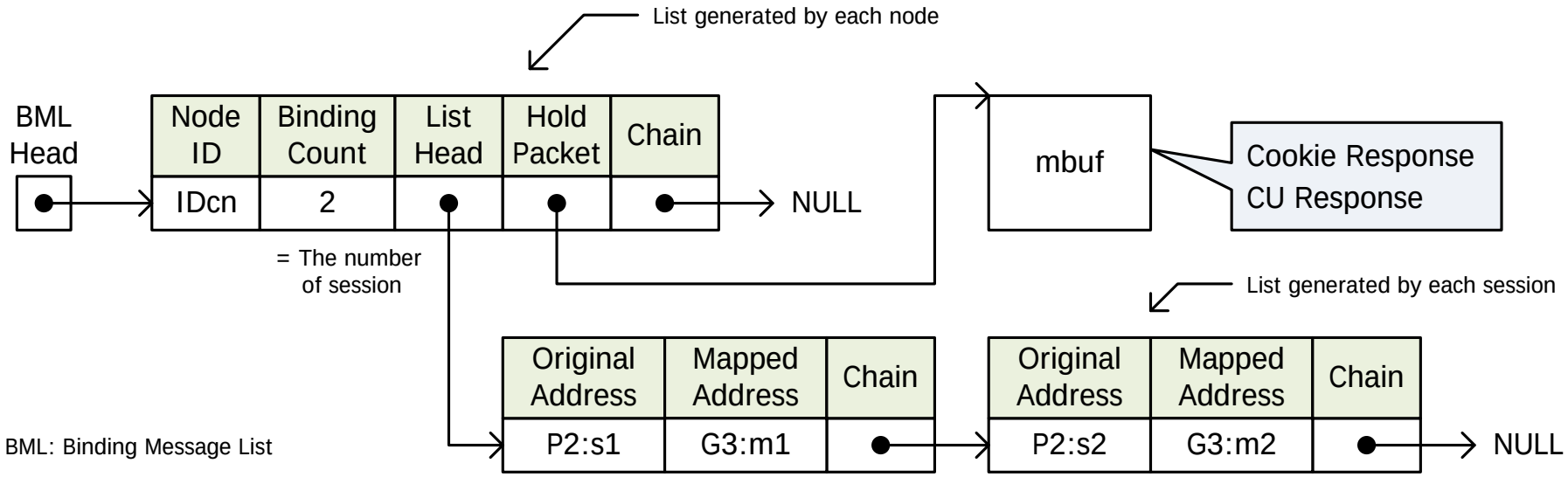
Type:		Code:		Flags:	
REQUEST	1	COOKIE	1	NAT-ON-PATH	0x01
RESPONSE	2	DH KEY	2	NAT-OFF-PATH	0x02
		CU	3	KEEP-ALIVE	0x04
		BINDING	4		

State:			
OK	1	NG_AUTH	4
NG_COOKIE	2	NG_NAT_EXIST	5
NG_NO_AK	3		

(a) Mobile PPC Header

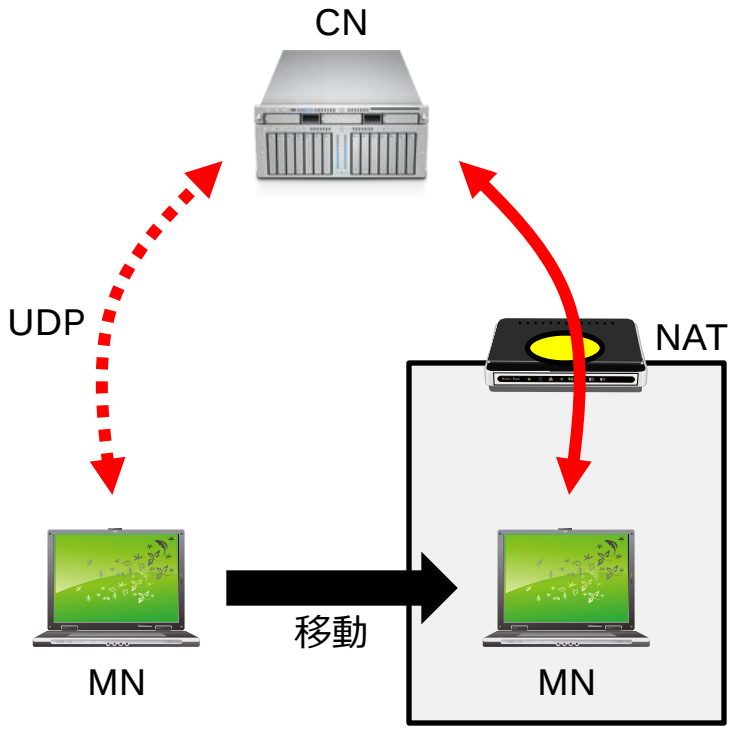
Binding Message List

- **BMLを新たに定義**
  - Bindingメッセージ情報の格納，トリガパケットの待避
    - **BML作成➔**Bindingメッセージの監視を開始
      - CU Responseに記載されたCIDの数をBinding Countとして設定
    - **BML削除➔**監視を終了
      - Binding Responseの受信ごとにBinding Countをデクリメント



動作確認

- Global to Privateの移動パターンを確認
  ➔UDP通信の継続を確認



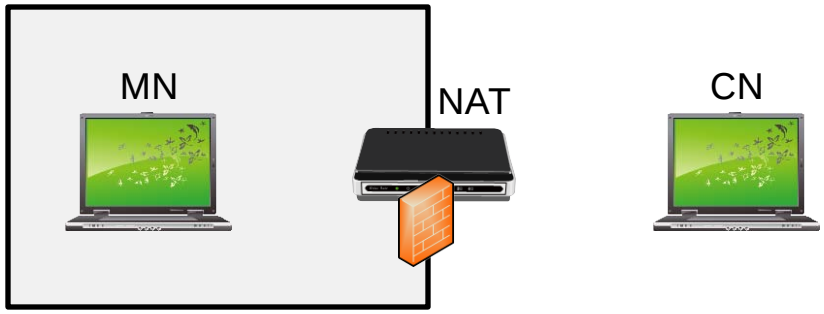
TCP通信は…

NATにファイアウォールが動作していない場合のみ，動作を確認

残された課題

- **TCPによるBinding処理とFire Wallの影響**
  - 多くのNATルータは**SPI（Stateful Packet Inspection）**機能を実装
    - OutboundパケットとInboundパケットの整合性を検査

    ➔正当な手順のTCPセッションと判断できない場合は，破棄



SYN [Out] ➔
SYN/ACK [in] ➔
ACK [Out] ➔

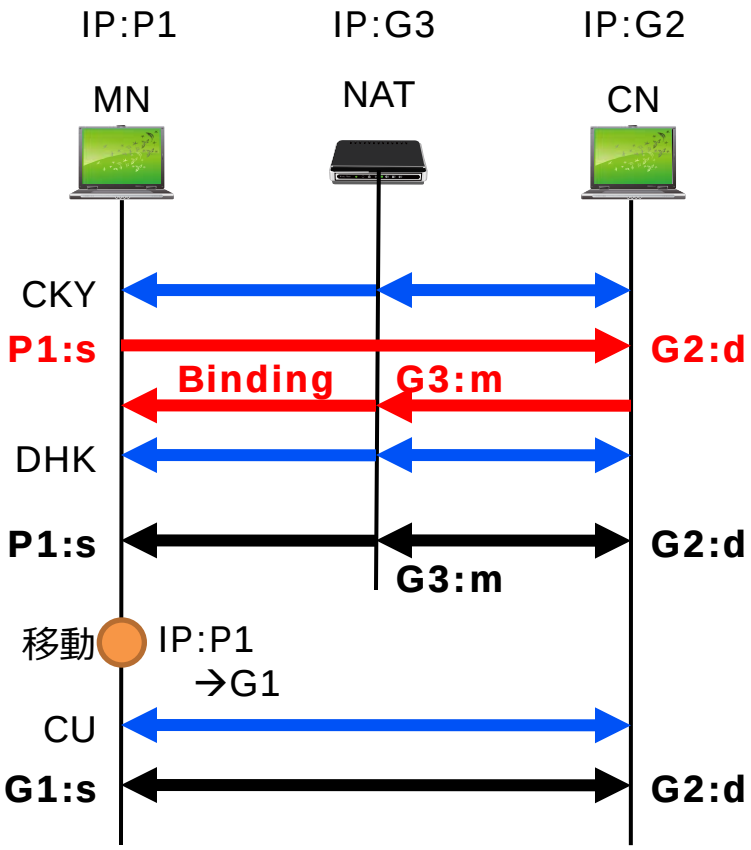
3-way handshake
➔Bindingメッセージを
1.5往復にする必要あり？

まとめ

- IPv4環境における移動透過性の実現
  - NAT Traversal問題を解決する必要がある
    - ➔Hole Punchingによりカプセル化なしで対応可能
- 提案方式を実装
  - 簡易的な動作確認を実施
    - ➔異なるアドレス空間をまたがった移動が可能
- 今後の課題
  - 実装を完了➔評価
  - 通信相手がプライベートネットワークにいる場合

付録

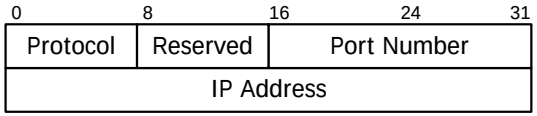
Private to Global



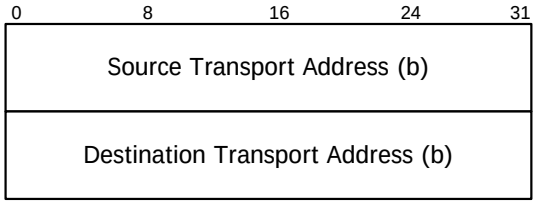
- **Cookie交換**
  - データ部＝P1，IPヘッダ＝G3
  - CNは**アドレスの不一致を検出**
    ➔Binding処理をMNに要求
- **Binding Req./Res.**
  - MAPPED-ADDR＝G3:m
- **CU Request**
  - Binding結果を反映
    - 移動前＝**G3:m**，移動後＝**G1:s**

制御メッセージフォーマットの拡張(2)

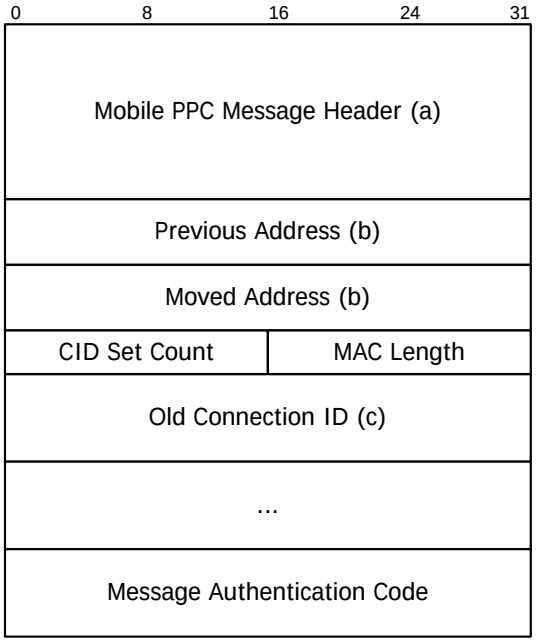
- トランスポートアドレス構造体を定義
  - IPアドレス，ポート番号，プロトコルを一括管理
    - コネクション識別子やCUメッセージにも反映



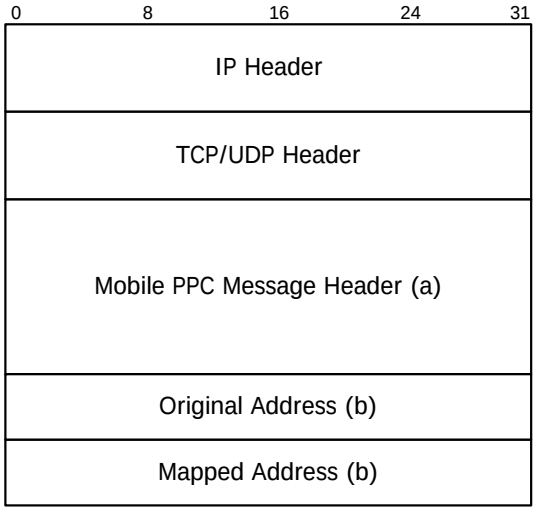
(b) Transport Address



(c) Connection ID



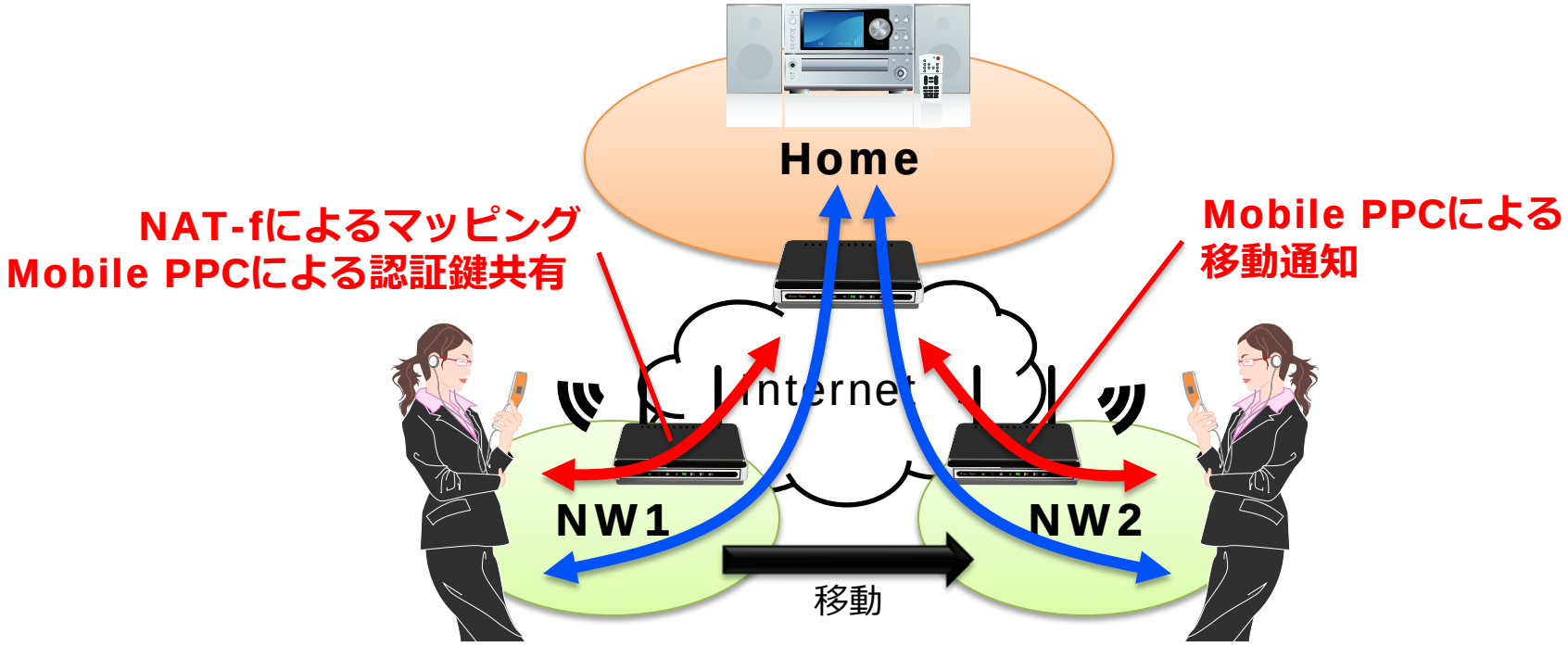
(d) CU Message



(e) Binding Message

CNがプライベートNWにいる場合 (1)

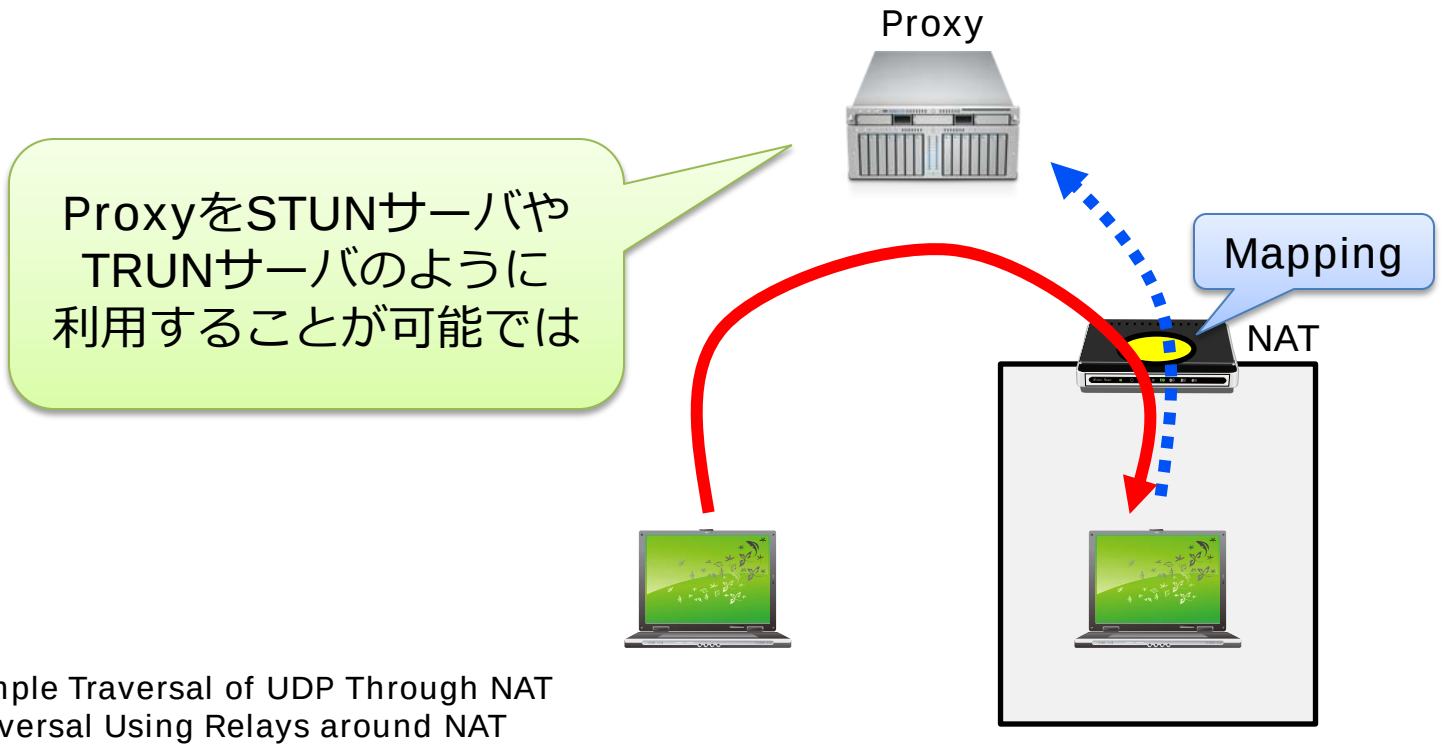
- **NAT-f (NAT-free protocol)を併用**
  - 通信開始時にNAT外側からダイナミックにマッピング生成



* 鈴木，渡邊：信学論(B)，Vol.J92-B, No.1, pp.109—121, Jan. 2009.

CNがプライベートNWにいる場合（2）

- **プロキシサーバを利用**
  - 通信相手がMobile PPC非対応の場合に利用する装置
  - プロキシ宛にHole Punching
    - ➔NAT外部からのリーチャビリティを確保



STUN: Simple Traversal of UDP Through NAT
TURN: Traversal Using Relays around NAT