平成22年度　修士論文

邦文題目

NATを跨る移動透過性を実現するNTMobileの提案

英文題目

Proposal of NTMobile that realizes mobility over NATs

情報工学専攻

(学籍番号: 093430031)

水谷　智大

提出日: 平成23年1月31日

名城大学大学院 理工学研究科 修士課程

内容要旨

近年のネットワーク環境では小型携帯端末や公衆無線網の普及により，移動しながら通信を行いたいという需要がある．しかし TCP/IP は移動通信を想定していなかったため，ノードは移動すると通信が切断されるという課題があった．この課題を解決する移動透過性の研究は盛んに行われてきたが，将来への IPv6 への移行を見越し，そのほとんどが IPv6 をベースにした技術であった．しかし IPv6 は IPv4 と互換性がないことから普及が進んでおらず，今後も IPv4 は使用され続けると考えられる．ところが IPv4 では，組織のネットワークをプライベートアドレスで構築するため外部から内部に通信を開始できないいわゆる NAT 越え問題が発生する．すなわち，IPv4 において移動透過性を実現するに当たり，NAT 越えも同時に実現できなければならない．

本論文では，NAT の存在に関わらず自由に通信を開始でき，かつあらゆる移動を行うことができる NTMobile（NAT Traversal with Mobile）を提案する．また本手法の実装の設計についても検討を行ったので報告する．

目 次

第1章　はじめに

インターネットにおいて，TCP/IP は通信インフラとして広く普及している．しかし近年のネットワーク環境は TCP/IP が当初想定していた状況を遥かに超え，様々な課題が明らかになっている．最大の課題は IPv4 グローバルアドレスの枯渇である．この課題への長期的対策として，IPv6 [1] が IETF により定義された．しかし IPv6 は IPv4 と互換性がなく，未だ本格的な普及に至っていない．今後 IPv6 への移行が進んだとしても全ての IPv4 機器を IPv6 機器へ変更するためには多大なコストが必要であり，当分の間 IPv4 が主流を占めるものと予想される．

短期的な解決策として，組織内ネットワークはプライベートアドレスで構成し，インターネットと組織内ネットワークの境界に NAT を設置することが一般的となっている．しかし NAT はプライベートネットワークをグローバルネットワークから隠蔽する性質を持つため，NAT 外部から内部に対して通信開始ができない NAT 越え問題が発生する．これは IPv4 通信の汎用性を損なう大きな要因となっており，NAT 越え問題の解決は重要な課題である．これまで，企業などの組織ではインターネットから組織内部への接続は許可されていないことが多く，NAT 越え問題は表面化しなかった．しかし近年では一般家庭でもネットワークが構築され，外出先から家庭内ネットワークにアクセスしたいという需要が出てきている．

NAT 越え技術には，大別すると NAT そのものに機能を加えるものと，NAT には手を加えず，通信を行う両エンドノードのみ，もしくは第三の機器の補助を受けて実現するものがある．NAT に改造を加えることで NAT 越えを実現する技術には 4+4 [2]，AVES [3]，UPnP [4]，NAT-PMP [5]，RS-IP [6, 7]，C-NATS [8]，IPNL [9]，NAT-f [10] などがある．これらの方式はパケットに対するカプセル処理や中継処理が不要なため，スループットが高いという特徴がある．しかし特殊な NAT を用意する必要があるため，接続できるプライベートネットワークが限定されるという課題がある．

NAT に手を加えずに実現する方式には Hole Punching [11] をベースにした Teredo [12]，第三の機器を中継する TURN [13] などがある．また両方の技術を組み合わせた方式として SIP [14] をベースにした ICE [15] がある．これらの方式では接続できるプライベートネットワークに制限はない．しかし Hole Punching はセッションの概念を持つ TCP では実現が難しく，ほぼ UDP に限られる．また中継機器を利用する手法ではスループットが低下するという課題がある．

TCP/IP において，次に大きな課題として挙げられるのは，ノードが通信中に移動した際に通信継続が難しい点が挙げられる．近年の小型携帯端末の普及や公衆無線ネットワー

ク環境の整備により，屋内・屋外を問わず，移動しながらインターネットを利用したいという需要が増加している．しかし TCP/IP では，通信識別子として使用される IP アドレスがネットワークに接続する場所に依存している．そのため，ノードが通信中にネットワークを移動すると通信識別子が変化し，移動前に行っていた通信が別の通信と見なされ，通信が切断されてしまうという課題がある．

この課題を解決する移動透過性の研究は，将来への IPv6 への移行を見越し，そのほとんどが IPv6 をベースにしている．しかし今後の IPv6 の展開を考慮すると，IPv4 における移動透過性技術も重要である．IPv4 における移動透過性技術には Mobile IP [16] や Mobile PPC [17] がある．Mobile IP では，第三の機器として，移動ノード（MN; Mobile Node）の位置の管理とパケットのカプセル化及び転送を行なう HA（Home Agent）を必要とする．そのため，スループットの低下が発生する他，HA に障害が起きると通信不能になる．また MN は移動後，通信相手ノード（CN; Correspondent Node）に送信するパケットの送信元アドレスを，移動先のネットワークアドレスと異なる IP アドレスにするため，通信経路上のルータで破棄される可能性がある [18].

Mobile PPC（Mobile Peer to Peer Communication）では通信を行う両エンドノードが Mobile PPC の機能を実装している必要があるが，MN の移動情報を両エンドノード間で直接通知するため，位置を管理する第三の機器を必要としない．また，両エンドノードがカーネルでパケットの IP アドレスを変換することで，上位アプリケーションに対して移動ノードの移動による通信識別子の変化を隠蔽するため，中継処理やカプセル処理を必要とせず，スループットの低下もほとんど見られない．これによりネットワークを流れるパケットの送信元 IP アドレスが MN の移動先のネットワークで取得したものであるため，ルータで破棄されることもない．このように Mobile PPC は Mobile IP の課題をほとんど解決できる．しかし，Mobile PPC は NAT の存在を考慮していなかったため，NAT を跨る移動に制限があるという弱点がある．

ユビキタスネットワークにおいて，IPv4 の環境での自由な通信開始と通信中の移動を実現するためには，NAT 越えと移動透過性を同時に実現しなければならない．そのような技術として，Mobile IP や Mobile PPC に NAT 越え技術を組み合わせた方式が研究されてきた．Mobile IP では UDP トンネルを使用する手法 [19] や NAT に独自機能を追加する手法 [20] などがある．文献 [19] では，通信開始時や移動時の NAT 越えも同時に実現するが，中継やカプセル処理によるスループットが低下するという課題がある．文献 [20] は通信開始時の NAT 越えを実現しているが，MN が移動できる範囲は同一のプライベートネットワーク内に限定され，NAT を跨った移動はできない．

Mobile PPC では Hole Punching を用いる手法 [21] や，NAT-f（NAT-free Protocol）を組み合わせる手法 [22] がある．文献 [21] では，MN が移動する際に CN がグローバルネットワークに存在しなければならない．また TCP の場合，NAT の SPI（Stateful Packet Inspection）機能でパケットが破棄される可能性がある．文献 [22] では，CN が NAT-f ルータ配下のプライベートネットワークのみに制限されるという課題がある．

本研究では，NAT の存在や NAT の SPI 機能に左右されることなく自由に通信を開始でき，また通信中に NAT を跨るあらゆる移動に対応できる，NTMobile（NAT Traversal with Mobile）を提案する．提案方式では，全ての通信を UDP トンネルで実行することにより，NAT 越えと移動透過性を同時に実現する．また，IP アドレスの通信識別情報と位置情報を明確に分離し，通信識別子に到達性のない仮想アドレスを割り当てることで，移動による IP アドレスの変化を上位ソフトウェアに対して隠蔽する．本方式を実現するため，両エンドノードに対して UDP トンネルの生成を指示する機能を DDNS（Dynamic DNS）[23] に持たせる．

本提案の実装について，UDP トンネルの生成方法の問い合わせやその応答，UDP トンネルの生成処理は，アプリケーションデーモンとして実装する．また通信パケットのアドレス変換処理と UDP カプセル処理は，カーネルモジュールとして実装する．

以降，2 章で NAT 越えと移動透過性を同時に実現する既存技術の概要とその課題を述べ，3 章で提案方式について述べる．4 章では提案方式の実装方法について述べ，最後に 5 章でまとめる．

第2章　NAT越えと移動透過性を同時に実現する既存技術とその課題

IPv4 において NAT 越えと移動透過性を同時に実現できる既存技術として，Mobile IP と Mobile PPC をベースにした方式がある．ここではそれらの概要と課題について述べる．以降の説明で使用する記号について以下に示す．

- $G*$　　；グローバルアドレス
- $P*$　　；プライベートアドレス
- $A:a$　　；IP アドレス A，ポート番号 a
- $A:a \rightarrow B:b$　；送信元 $A:a$ から宛先 $B:b$ のパケット
- $A:a \leftrightarrow B:b$　；$A:a$ と $B:b$ との通信
- $A:a \Leftrightarrow B:b$　；$A:a$ と $B:b$ のアドレス変換

2.1　Mobile IP ベースの方式

Mobile IP では通信を行うエンドノードの他に，第三の機器として HA を導入する．HA は MN のホームネットワークに置かれ，MN の位置情報の把握や MN へのパケットの転送を行う．また MN はホームネットワークで取得する HoA（Home Address）と移動先のネットワークで取得する CoA（Care-of Address）の二つの IP アドレスを持つ．HoA は移動しても変化しない IP アドレスであり，両エンドノードが常に HoA を MN として認識して通信を行うことで，MN の移動後の通信を継続する．図 2.1 に Mobile IP の動作概要を示す．MN は移動先で G_{CoA} を取得すると，MN の HoA "G_{HoA}"と CoA "G_{CoA}"を記載した Binding Update を HA に送信する．HA はこれを受信すると G_{HoA} と G_{CoA} の対応関係を示すエントリを生成した後，G_{HoA} 宛のパケットを受信するようにホームネットワーク内に ARP を通知する．これにより以後の通信では，CN が G_{HoA} 宛にパケットを送信すると，このパケットは HA に送り届けられる．

HA は G_{HoA} 宛のパケットを受信すると，アドレスの対応関係からこのパケットに対して G_{CoA} 宛の IP-in-IP カプセルを形成し，移動先の MN に転送する．MN はこのパケットを受け取るとカプセルを解除し，G_{HoA} 宛のパケットを取り出す．また MN がパケットを送信する際は，送信元のアドレスを G_{HoA} にして直接 CN に送信する．このように HoA を通信識別子として用い，CoA を位置情報として用いることで，CN と MN は，MN の移動に関わらず常に MN を G_{HoA} として認識したまま通信を行うことができる．

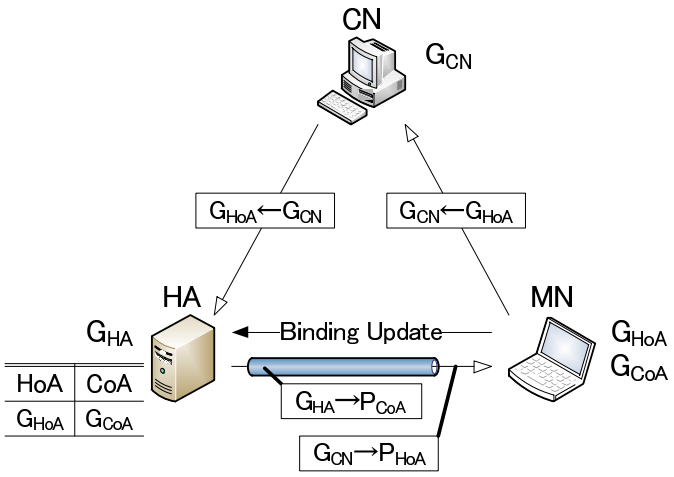

図 2.1　Mobile IP の動作概要

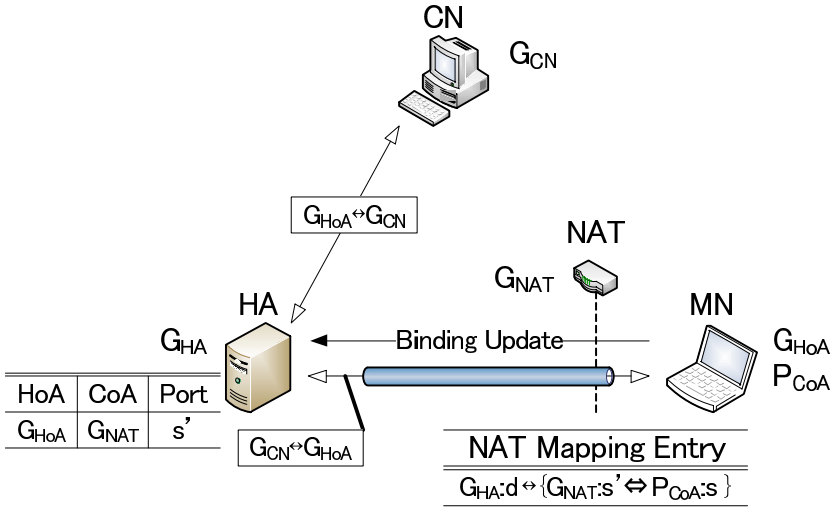

図 2.2　UDP トンネルを利用した Mobile IP の動作概要

Mobile IP では NAT 越え問題に対応するため，UDP トンネルを利用する方式 [19] や特殊な NAT を導入する方式 [20] が提案されている．UDP トンネルを利用した Mobile IP を図 2.2 に示す．この方式では MN が NAT 配下に移動して P_{CoA} を取得すると，MN は HoA ”G_{HoA}”を記載した UDP ベースの Binding Update を HA に送信する．この時，NAT のマッピングテーブルには以下のマッピングエントリが生成される．

$$G_{HA} : d \leftrightarrow \{G_{NAT} : s' \Leftrightarrow P_{CoA} : s\}$$

ここで s' は，NAT が $G_{HA} : d \leftrightarrow P_{CoA} : s$ の通信に対して割り当てたポート番号である．

HA は Binding Update を受信すると，MN の CoA として G_{NAT}，また UDP トンネルのポート番号 s' をエントリに登録する．その後，G_{HoA} 宛のパケットを受信すると，$G_{NAT} : s'$ 宛の UDP トンネルを使用して転送する．このパケットは NAT により $G_{NAT} : s'$ から $P_{CoA} : s$ に変換され，MN まで転送される．また MN が CN に パケットを送信する場合も同じ UDP

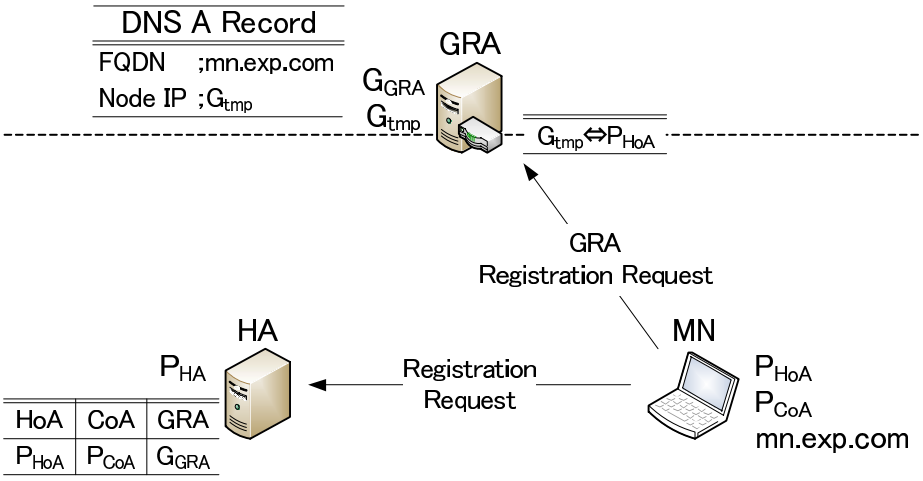

図 2.3　GRA を用いた Mobile IP の登録処理

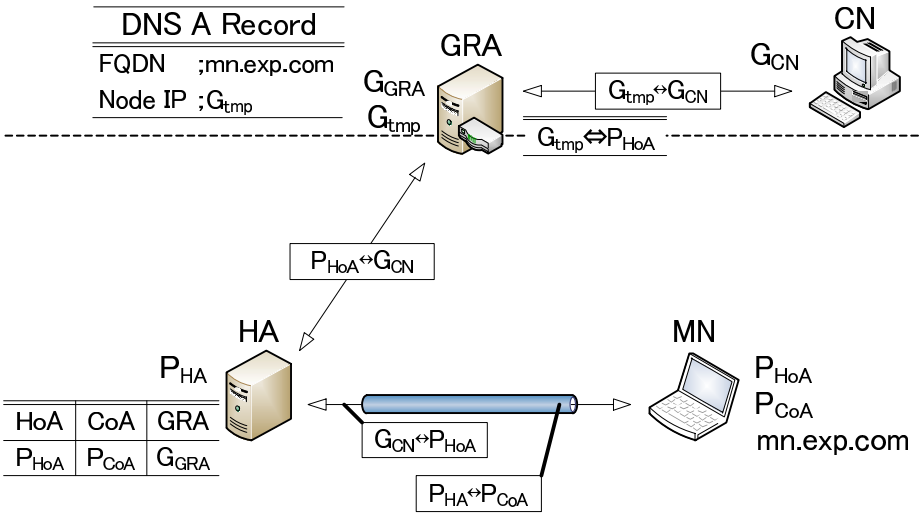

図 2.4　GRA を用いた Mobile IP の転送処理

トンネルを使用して HA へ送信し，HA は UDP カプセルを解除して転送する．これによりパケットは NAT に影響されなくなるため，MN は NAT 配下に移動できるようになる．また，CN がプライベートネットワーク内の MN に対して通信を開始することができる．しかし，パケットが全て HA を経由して UDP カプセル処理が行われるため，スループットの更なる低下が課題となる．

後者の方式では，MN のホームネットワークや移動先のネットワークに HA と GRA（Global Roaming Agent）と呼ぶ特殊な NAT 機器を導入することにより，プライベートネットワーク内の MN に対して通信開始可能にする．図 2.3 に GRA を用いた Mobile IP の登録処理の概要を，図 2.4 に転送処理の概要を示す．

MN は Mobile IP の Binding Update に当たる Registration Request を HA に送信し，自身の HoA と CoA である P_{HoA}，P_{CoA} と，GRA の IP アドレス G_{GRA} を登録する．この時，

HA は通常の Mobile IP と同様，P_{HoA} 宛のパケットを受信するようにホームネットワーク内に ARP を通知する．また MN は自身のホスト名 "*mn.exp.com*"と HoA "P_{HoA}"を記載した GRA Registration Request を GRA に送信する．GRA は *mn.exp.com* に対して一時的なグローバルアドレス G_{tmp} を割り当て，P_{HoA} に対して以下の変換テーブルエントリを生成する．

$$G_{tmp} \Leftrightarrow P_{HoA}$$

また，GRA は DNS の機能を有しており，*mn.exp.com* に対する A レコードとして G_{tmp} を登録する．

その後 CN は MN と通信を開始するため，*mn.exp.com* の名前解決を行うと GRA から G_{tmp} を受け取り，G_{tmp} 宛のパケットを送信する．GRA はこのパケットを受信すると，変換テーブルに従って，宛先 IP アドレスを G_{tmp} から P_{HoA} に変換して転送する．転送されたパケットは HA に届けられ，HA は通常の Mobile IP と同様，P_{CoA} 宛の IP-in-IP トンネルを用いて MN に転送する．MN が送信する場合は，CN から送信された経路と同一の経路を用いて送信する．

以上の処理により MN に対する通信開始時の NAT 越えを実現している．しかし，MN はプライベートネットワーク毎に異なる HoA を使用して通信を行うため，通信を継続したまま異なるプライベートネットワークへ移動することはできず，その範囲は同一の GRA 配下のプライベートネットワークのみに限定される．

2.2 Mobile PPC をベースにした方式

Mobile PPC は，IPv4 において第三の装置の補助を必要とせず，エンドエンドで移動透過性を実現するプロトコルである．Mobile PPC では各エンドノードがカーネルの IP 層に CIT（Connection ID Table）と呼ぶアドレス変換テーブルを持ち，両エンドノードがこれを参照して全てのアプリケーションパケットの IP アドレスを変換することにより，上位アプリケーションに対してアドレスの変化を隠蔽し，かつパケットを正しくルーティングして通信を継続することができる．Mobile PPC は，通信開始時のアドレス解決と，通信中の移動に係わるアドレス解決を明確に分離し，前者を DDNS により実現し，後者を Mobile PPC が実現する．

Mobile PPC の動作を図 2.5 に示す．Mobile PPC では両エンドノード共に移動できるため CN, MN といった区別は存在しないが, 便宜上, 図 2.5 で移動するエンドノードを MN, その通信相手ノードを CN とする．CN と MN は既に DDNS を用いて，$G_{CN} : s \leftrightarrow G_{MN} : d$ の通信を開始しているものとする．通信中に MN が移動して新しく IP アドレス $G_{MN'}$ を取得すると MN は CN との間で MN の移動前後の IP アドレス G_{MN}，$G_{MN'}$ を記載した CU（CIT Update） Request/ Response を交換し，両エンドノードは以下のような CIT エントリを生成する．

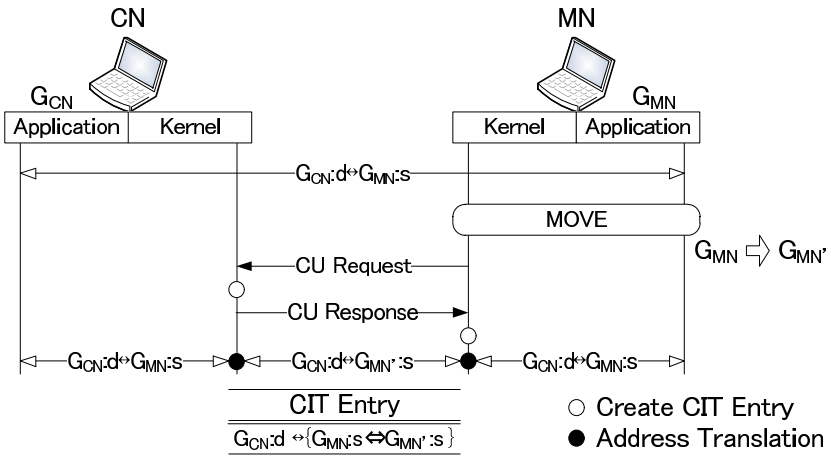

図 2.5　Mobile PPC の動作概要

$$G_{CN} : s \leftrightarrow \{G_{MN} : d \Leftrightarrow G_{MN'} : d\}$$

以後，MN が CN にアプリケーションパケットを送信する場合は IP 層で CIT を参照し，パケットの送信元アドレスを G_{MN} から $G_{MN'}$ に変換して送信し，CN はこれを受信すると，逆に $G_{MN'}$ から G_{MN} に変換して上位アプリケーションに渡す．CN が MN にパケットを送信する場合は逆の処理を行う．以上の処理によりパケットは正しくルーティングされ，かつ両エンドノードの上位アプリケーションは MN の IP アドレスが変化したことに気付くことなく通信を継続できる．

Mobile PPC では NAT 越え問題に対応するため，Hole Punching を組み合わせる方式と，NAT-f を組み合わせる方式が提案されている．

図 2.6 に Hole Punching を用いた Mobile PPC の動作概要を示す．MN は移動して新しく IP アドレスを取得すると CN との間で CU Request/ Response を交換するが，この時 CN は CU Request に記載されている MN の移動後の IP アドレス $P_{MN'}$ と IP ヘッダの IP アドレス G_{NAT} が異なることを検出すると，MN が NAT 配下に移動したと判断する．その場合，CN は MN に Hole Punching に当たる Binding 処理を行うよう要求するフラグを付けた CU Response を返す．MN は要求に従い，CN との通信で使用しているプロトコルをベースにした Binding Request/ Response を CN との間で交換する．これにより NAT のマッピングテーブルには以下のマッピングエントリが生成される．

$$G_{CN} : d \leftrightarrow \{G_{NAT} : s' \Leftrightarrow G_{MN'} : s\}$$

その後，MN は CN との間で再度 CU Request/ Response を交換し，CN，MN はそれぞれ以下のような CIT エントリを生成する．

$$\text{CN;}\quad G_{CN} : d \leftrightarrow \{G_{MN} : s \Leftrightarrow G_{NAT} : s'\}$$
$$\text{MN;}\quad G_{CN} : d \leftrightarrow \{G_{MN'} : s \Leftrightarrow G_{MN} : s\}$$

以上の動作により，マッピングエントリと CIT エントリが対応付けられる．以後は通常の Mobile PPC と同様，CIT テーブルを参照したアドレス変換を行うことで通信を継続する．

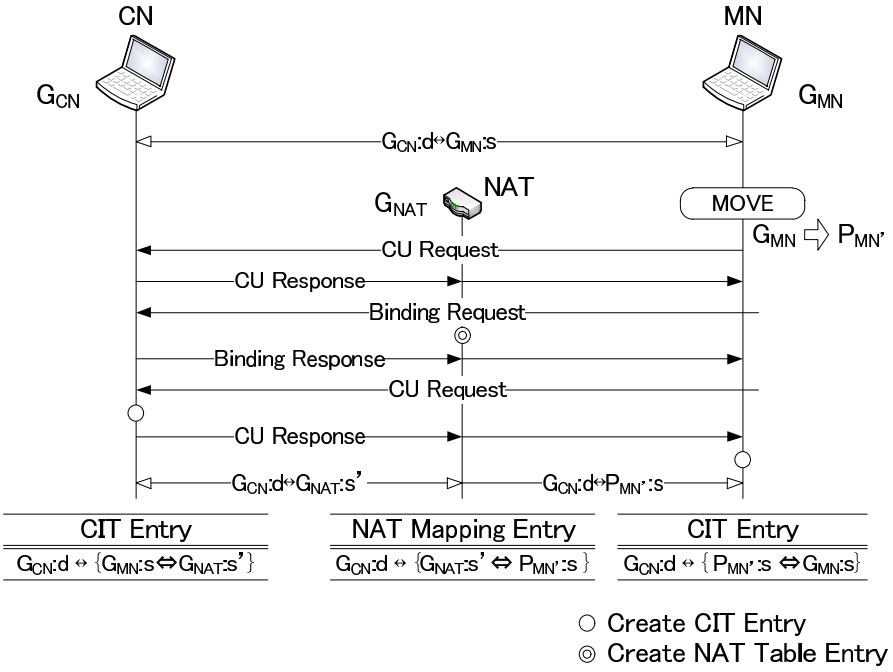

図 2.6　Hole Punching を用いた Mobile PPC の動作概要

近年の NAT には，SPI と呼ばれる TCP シーケンス番号等の通信の整合性をチェックするフィルタリング手法が搭載されていることが多い．このため，Binding Update によって形成した TCP セッションのシーケンス番号と移動前に行っていた通信の TCP セッションのシーケンス番号が異なると，通信経路上の NAT が SPI 機能を有する場合，TCP パケットが破棄される可能性がある．

またこの方式では MN が移動する際に CN が NAT 配下に存在すると，MN からの CU Request が CN に到達せず，Binding 処理を開始できない．したがって MN の移動時，CN はグローバルネットワークに存在しなければならない．更に通信開始時の NAT 越えは想定しておらず，プライベートネットワーク内の CN に対して通信開始するには別の技術との組み合わせが必要である．

後者の方式では NAT-f ルータに Mobile PPC の機能を，MN に Mobile PPC と NAT-f 機能を導入する．図 2.7 に，NAT-f を用いた Mobile PPC の動作概要を示す．MN は IP アドレス G_{MN} を持ち，CN の IP アドレス P_{CN} に対し，ポート番号 s から d に通信開始する．

MN はまず通信開始時，NAT-f ルータとの間で NAT-f ネゴシエーションを行う．これにより，NAT-f ルータのマッピングテーブルには以下のマッピングエントリを生成する．

$$G_{MN} : s \leftrightarrow \{G_{NAT\,f} : d' \Leftrightarrow P_{CN} : d\}$$

これにより MN が送信する $G_{NAT\,f} : d'$ 宛のパケットは $P_{CN} : d$ に届けられ，CN に対して通信開始できる．

その後 MN が移動して $G_{MN'}$ を取得すると，NAT-f ルータとの間で CU Request/ Response を交換する．これにより MN，NAT-f ルータそれぞれに，以下の CIT エントリが生成さ

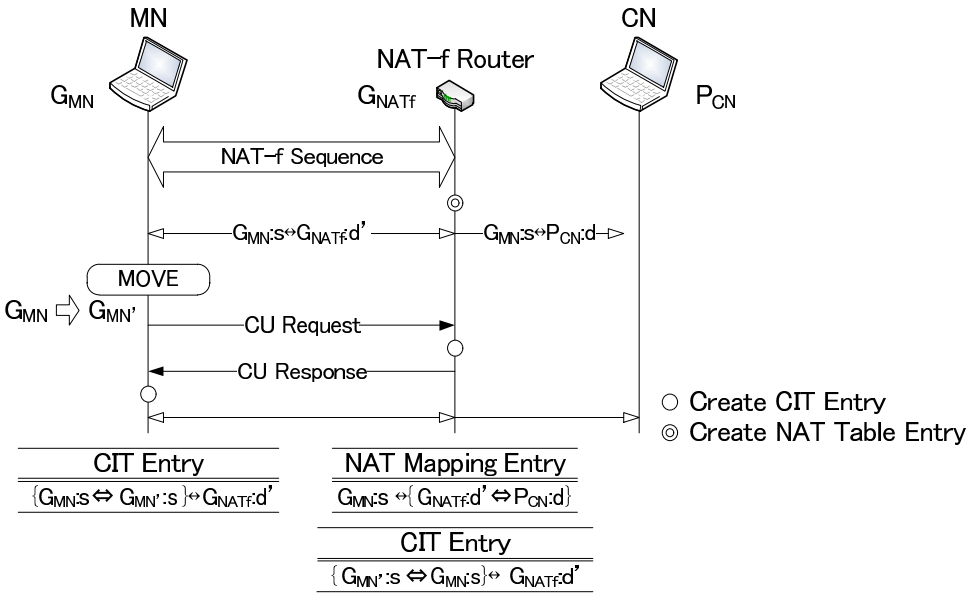

図 2.7　NAT-f を用いた Mobile PPC の動作概要

れる．

$$\text{MN};\quad \{G_{MN} : s \Leftrightarrow G_{MN'} : s\} \leftrightarrow G_{NAT\,f} : d'$$
$$\text{NAT-f ルータ};\quad \{G_{MN'} : s \Leftrightarrow G_{MN} : s\} \leftrightarrow G_{NAT\,f} : d'$$

以後，MN は CN にアプリケーションパケットを送信する際，CIT 参照してパケットの送信元アドレスを $G_{MN} : s$ から $G_{MN'} : s$ に変換する．NAT-f ルータはこれを受信すると，CIT を参照してパケットの送信元アドレスを $G_{MN'} : s$ から $G_{MN} : s$ に，更にマッピングテーブルを参照してパケットの宛先アドレスを $G_{NAT\,f} : d'$ から $P_{CN} : d$ に変換して CN に転送する．CN が送信する場合も，NAT-f ルータと MN で同様の変換処理が行われる．

しかし CN の移動は想定しておらず，接続する場所も NAT-f ルータ配下でなければならない．また MN が NAT 配下に移動してしまうと，その NAT によるアドレス変換により CIT の対応関係が崩れてしまう．そのため MN の移動範囲もグローバルネットワーク内のみに制限される．

第3章　提案方式

3.1　要求仕様

IPv4 ネットワークでは図 3.1 に示すような NAT を含む通信の構成が想定され，このような環境下でエンドノードが自由に移動できなければならない．即ち，移動透過性と NAT 越えを同時に実現することが必要である．

このとき，MN は様々なネットワークを移動することが想定されるため，NAT に改造を加えることは望ましくない．このとき NAT には SPI 機能が搭載されている可能性もあるが，このような場合にも対応できる必要がある．通信形態としては，両エンドノードが異なるプライベートネットワークに存在することもあり得る．このとき取得するプライベートアドレスは重複する可能性があるが，この場合にも対応できることが望ましい．更に，上位アプリケーションや使用するプロトコルの制限は望ましくない．様々な相手と通信を行なうことを考慮すると，MN が一般のサーバとの通信中にも移動できることが望ましい．

本論文では，これら全ての要求を満たす通信の実現を目標とする．

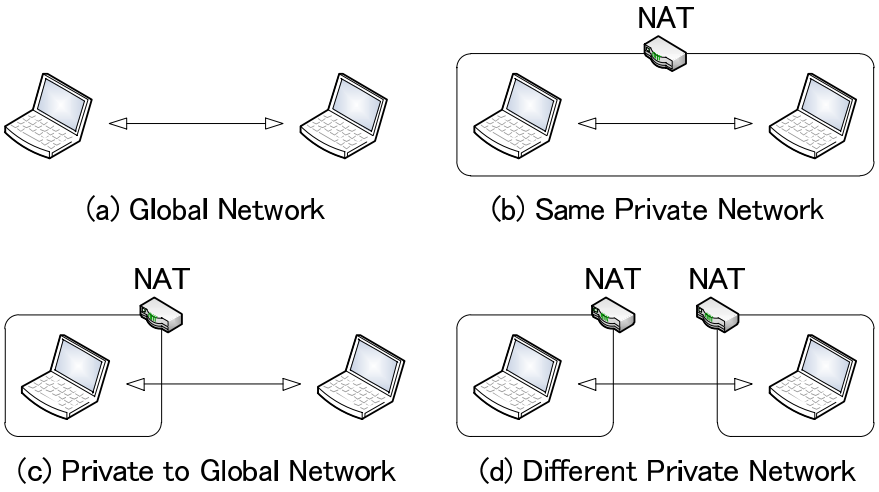

図 3.1　通信の構成

3.2 実現方法の概要

本提案では両エンドノード間で UDP トンネルを生成し，これを用いて通信を行なう．これにより上位アプリケーションや使用するプロトコルの制限，NAT の SPI を回避することができる．このとき，UDP トンネルの生成は UDP Hole Punching により行い，図 3.1 (c) のようにエンドノードのどちらか一方のみが NAT 配下に存在する場合には，NAT 配下のエンドノード側からこれを開始する．(d) のように両エンドノードがそれぞれ異なる NAT 配下に存在する場合には，確実な通信を行うために両エンドノードが第三の機器に対して UDP Hole Punching を行い，UDP トンネルを用いた中継通信を行う．また CN が一般サーバの場合は第三の機器をプロキシサーバとして使用し，MN はプロキシサーバとの間で UDP トンネルを形成する．

本提案ではこれらの通信を実現するため，DDNS に対し，UDP トンネル生成方法を両エンドノードに指示する機能を追加し，これを DS（Direction Server）と呼ぶ．また中継通信を行うための第三の機器として RS（Relay Server）を導入する．ただし RS は，両エンドノードがそれぞれ異なるプライベートネットワークに存在する場合と，CN が一般サーバの場合にのみ使用され，それ以外の場合は RS がなくても両エンドノードは通信可能である．なお，RS は DS と同一のサーバに機能を実装しても，負荷を分散するために別の機器で構成しても構わない．したがって，既存のネットワーク設備に対して改造を行う必要はない．

また，本方式ではノードの移動による IP アドレスの変化に対応するため，IP アドレスの通信識別情報と位置情報を明確に分離し，各エンドノードの内部でのみ有効な到達性のない仮想アドレスを導入して，通信識別情報にこれを割り当てる．また位置情報には従来と同じく，接続するネットワークで取得する IP アドレスを使用し，これを仮想アドレスに対し物理アドレスと呼ぶ．上位アプリケーションは通信を仮想アドレスで認識し，各エンドノードはアプリケーションパケットの送受信時に仮想アドレスと物理アドレスを変換する．これにより，図 3.1 (d) で通信する場合に両エンドノードが取得したプライベートアドレスが重複しても，通信を行うことができる．エンドノードが移動先で新しく IP アドレスを取得しても，通信を識別する仮想アドレスは変化することはなく，通信を継続することができる．

なお，各エンドノードと，そのエンドノードが依存する DS は共通鍵を予め保持していることを前提とし，この共通鍵で MAC（Message Authentication Code）の計算処理や制御パケットの暗号化処理を行うことで，エンドノードが DS とやり取りする NTM メッセージの機密性及び完全性と真正性を保証する．

3.3 NTM の詳細

以下では，通信開始時の処理と，エンドノードが移動した後の処理に分けて本提案方式の詳細を説明する．なお，本方式では両エンドノードが共に移動可能であるため，エ

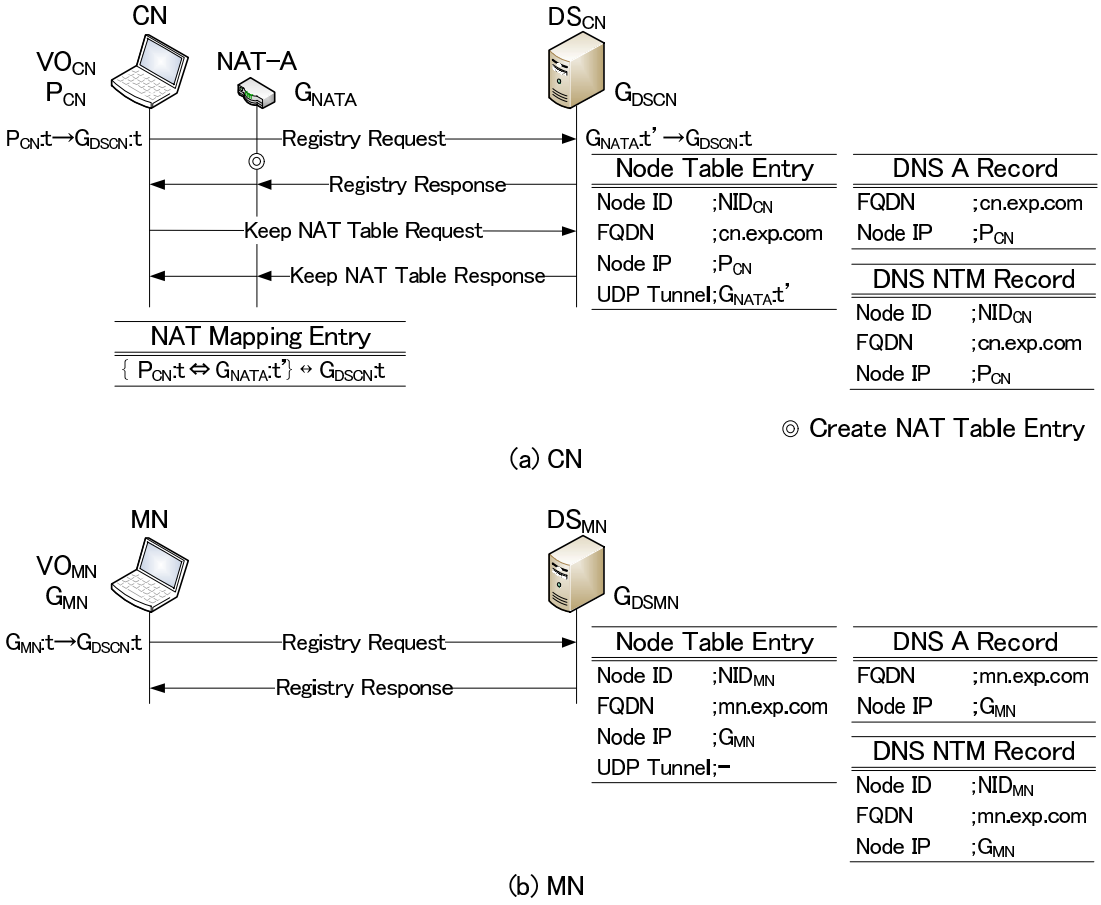

図 3.2　通信開始時の登録処理

ンドノードに区別はないが，便宜上，移動を行うエンドノードを MN，その通信相手ノードを CN とする．以降の説明で使用する記号について，以下に示す．

- $VO*$　　；自分自身のエンドノードを示す仮想アドレス
- $VC*$　　；通信相手のエンドノードを示す仮想アドレス

3.3.1　通信開始時の処理

(1)　エンドノードの情報登録

以下では通信開始時の処理として，図 3.1 の (c) の構成でグローバルネットワーク側の MN がプライベートネットワーク内の CN に通信を開始する場合を説明する．図 3.2 に通信開始時の登録処理を示す．

CN，MN が依存する DS をそれぞれ DS_{CN}，DS_{MN} とし，これらは IP アドレス G_{DSCN}，G_{DSMN} を持つ．また各 DS は各エンドノードに関する情報を管理する Node Table を持ち，DNS 機能には新たなレコードとして NTM レコードを定義する．NTM レコードはエンド

ノードに関する情報として，エンドノードを識別する Node ID[1] とエンドノードの物理アドレス，属する NAT の IP アドレスで構成される．

CN は IP アドレス G_{NATA} を持つ NAT-A 配下に接続して，物理アドレス P_{CN} と仮想アドレス VO_{CN} を持ち，MN は物理アドレス G_{MN} と仮想アドレス VO_{MN} を持つ．また各エンドノードは，仮想アドレスと物理アドレスの対応関係を保持した VRT（Virtual Real Translation），通信相手に関する情報を管理する Tunnel Table を持つ．

CN はネットワークに接続すると，自身の UDP ポート番号 t から DS_{CN} の UDP ポート番号 t に Registry Request を送信し，CN の Node ID ”NID_{CN}”，FQDN ”*cn.exp.com*”，物理アドレス ”P_{CN}”を報告する．この時，NAT-A はマッピングエントリに以下のようなマッピングエントリを生成する．

$$\{P_{CN} : t \Leftrightarrow G_{NATA} : t'\} \leftrightarrow G_{DSCN} : t$$

DS_{CN} はこれを受信すると，これらの情報に加え，IP/UDP ヘッダから NAT-A の外側の IP アドレス ”G_{NATA}”と NAT-A に割り当てられたマッピングエントリ ”$G_{NATA} : t'$”の Node Table エントリを生成する．また DS_{CN} はこれと同時に，自身の DNS サーバ機能に対して CN の A レコードとして IP アドレス ”G_{NATA}”を，NTM レコードとして CN の Node ID ”NID_{CN}”，FQDN ”*cn.exp.com*”，物理アドレス ”P_{CN}”を登録する．その後 CN は NAT-A のマッピングエントリを維持するため，定期的に $P_{CN} : t \rightarrow G_{DSCN} : t$，$G_{NATA} : t' \rightarrow G_{DSCN} : t$ の Keep NAT Table Request/ Response を交換する．

MN も同様，Registry Request により自身の FQDN ”*mn.exp.com*”，IP アドレス ”G_{MN}”の DS_{MN} の Node Table エントリの生成と，DNS サーバ機能への A レコード，NTM レコードの登録を行う．なお MN はグローバルネットワークに存在しているため，NAT の情報は登録されない．

(2) UDP トンネル生成ネゴシエーション

MN が CN に通信開始する場合の UDP トンネル生成ネゴシエーション処理を図 3.3 に示す．MN の上位アプリケーションはまず，CN の FQDN から IP アドレスを取得するため，”*cn.exp.com*”を記載した A レコードの DNS クエリをプライマリ DNS サーバに送信する．この DNS クエリは通常の DNS 問い合わせにより DS_{CN} まで送られ，DS_{CN} は A レコードの DNS レスポンスとして，CN が属する NAT である NAT-A の IP アドレス G_{NATA} を応答する．なお，この DNS レスポンスには追加セクションとして DS_{CN} の IP アドレス G_{DSCN} が記載されている．MN はこれを受け取ると，カーネルモジュールにて一時的に退避する．

その後 MN は，以後の UDP トンネル生成ネゴシエーション処理に必要な CN に関する情報を取得するため，CN の FQDN ”*cn.exp.com*”を記載した NTM レコードの DNS クエリを，A レコードの DNS レスポンスの追加セクションにより取得した G_{DSCN} 宛に送信す

[1] UUID（Universally Unique ID）[24] で構成し，重複しないことが保証される．

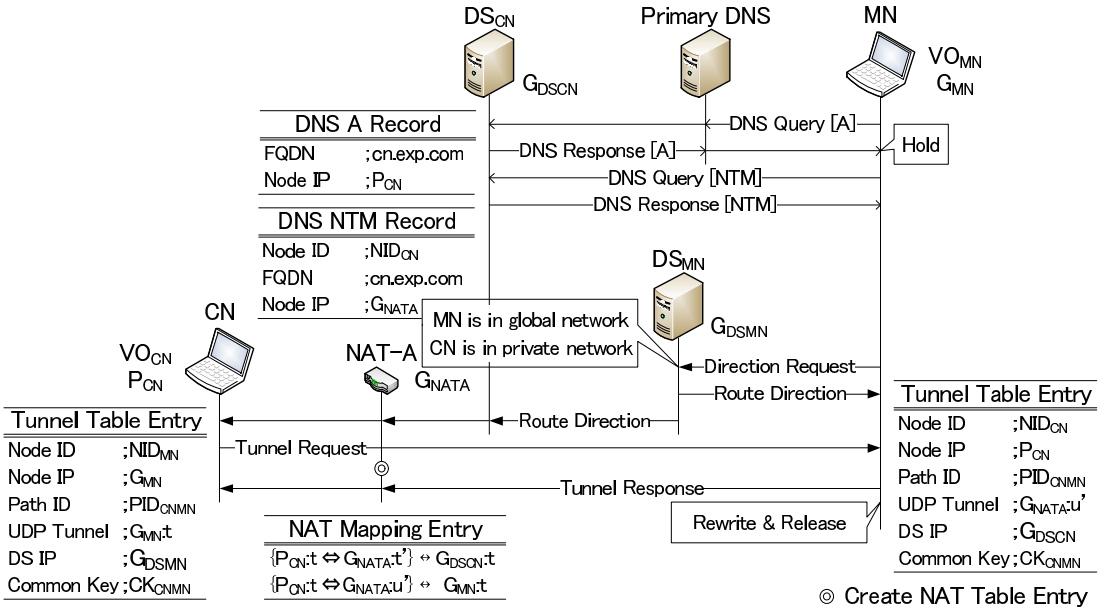

図 3.3　MN が CN に通信開始する場合の UDP トンネル生成ネゴシエーション処理

る．DS_{CN} はこれに対し，CN の Node ID ”NID_{CN}”，属する NAT の IP アドレス ”G_{NATA}”，物理アドレス ”P_{CN}”を記載した NTM レコードの DNS レスポンスを返す．

MN はこの NTM レコードの DNS レスポンスを受け取ると，CN との通信を一意に識別する Path ID ”PID_{CNMN}”と，CN との通信に使用する共通鍵 ”CK_{CNMN}”を生成し，これらに加え，受け取った NTM レコードに記載された CN に関する情報（Node ID ”NID_{CN}”，物理アドレス ”P_{CN}”）と自身の情報（Node ID ”NID_{MN}”，物理アドレス ”G_{MN}”），及び DS_{CN} の IP アドレス ”G_{DSCN}”を記載した Direction Request を DS_{MN} に送信する．DS_{MN} は Direction Request の送信元 IP アドレスと MN の物理アドレス G_{MN} を比較し，これらが同一であることから MN がグローバルネットワークに存在すると判断する．また，同様に Direction Request 内の CN の物理アドレス G_{CN} と NAT の IP アドレス G_{NATA} を比較し，これらが異なることから CN がプライベートネットワークに存在すると判断する．

DS_{MN} はこの判断結果から MN の Node ID ”NID_{MN}”，物理アドレス ”G_{MN}”，DS_{MN} の IP アドレス ”G_{DSMN}”，MN との通信を示す Path ID ”PID_{CMMN}”，MN との通信に使用する共通鍵 ”CK_{CNMN}”，Tunnel Request の送信先 IP アドレス ”G_{MN}”と Tunnel Request の送信を示すコードを記載した Route Direction を CN に送信し，MN に Tunnel Request を送信するように指示する．ただし，DS_{MN} は CN に直接パケットを送信する通信路を持たないため，この Route Direction は一度 DS_{CN} に送られ，DS_{CN} が CN からの情報登録処理の際に生成されたマッピングエントリを参照して CN に転送する．また同様に CN の Node ID ”NID_{CN}”と物理アドレス ”P_{CN}”，DS_{CN} の IP アドレス ”G_{DSCN}”，CN の属する NAT の IP アドレス ”G_{NATA}”，CN との通信を示す Path ID ”PID_{CMMN}”，Tunnel Request の受信待ちを示すコードを記載した Route Direction を MN に送信し，CN からの Tunnel Request を待つように指示する．

CN は Route Direction を受信すると，MN に関する情報（Node ID "NID_{MN}"，物理アドレス "G_{MN}"，DS_{MN} の IP アドレス "G_{DSMN}"，MN との通信を示す Path ID "PID_{CMMN}"，MN との通信に使用する共通鍵 "CK_{CNMN}"，UDP トンネル情報 "$G_{MN} : t$"）の Tunnel Table エントリを生成し，また VRT Table に以下のようなエントリを生成する．

$$\{VO_{CN} \Leftrightarrow P_{CN}\} \leftrightarrow \{VC_{MN} \Leftrightarrow G_{MN}\}$$

その後，指示に従って MN との通信を示す Path ID "PID_{CNMN}"を記載した Tunnel Request を G_{MN} に送信する．また MN も同様に，Route Direction を受信すると CN に関する情報（Node ID "NID_{CN}"，物理アドレス "P_{CN}"，CN の属する NAT の IP アドレス "G_{NATA}"，DS_{CN} の IP アドレス "G_{DSCN}"，CN との通信を示す Path ID "PID_{CMMN}"，CN との通信に使用する共通鍵 "CK_{CNMN}"）の Tunnel Table エントリを生成し，VRT Table に以下のようなエントリを生成する．

$$\{P_{CN} \Leftrightarrow VC_{CN}\} \leftrightarrow \{G_{MN} \Leftrightarrow VO_{MN}\}$$

その後は指示に従って Tunnel Request の受信待ちを行う．

その後 MN は，Tunnel Request を受信すると CN に関する Tunnel Table エントリに UDP トンネル情報 "$G_{NATA} : u'$"を追加登録し，一時退避していた A レコードの DNS レスポンスを G_{NATA} から VC_{CN} に書き換えた後，開放して上位アプリケーションに渡す．以上の処理により，MN と CN との間に UDP トンネルが生成され，以後，MN と CN はこの UDP トンネルを使用して通信を行う．

(3) アプリケーションデータの通信

MN は CN との間の UDP トンネルの生成処理と互いの情報の交換が完了すると，アプリケーションパケットの送受信を開始する．MN が CN との間で送受信するパケットの変換・カプセル処理を図 3.4 に示す．

MN の上位アプリケーションは CN にパケットを送信するため，$VO_{MN} : s \rightarrow VC_{CN} : d$ のパケットを生成する．MN はこのパケットをカーネルモジュールにて VRT Table を参照し，アドレスを $G_{MN} : s \rightarrow P_{CN} : d$ に変換した後，$G_{MN} : t \rightarrow G_{NATA} : u'$ でカプセル化して送信する．

送信されたパケットは NAT-A に送り届けられ, NAT-A はマッピングテーブルを参照して，このパケットのアドレスを $G_{MN} : t \rightarrow P_{CN} : t$ に変換して転送する．CN はこのパケットを受信すると，カーネルモジュールにて UDP カプセルを解除し，$G_{MN} : s \rightarrow P_{CN} : d$ のパケットを取り出す．その後 Tunnel Table を参照し，このパケットのアドレスを $VC_{MN} : s \rightarrow VO_{CN} : d$ に変換して上位アプリケーションに渡す．

CN が MN に送信する場合も同様，アプリケーションパケットの IP アドレスを $VO_{CN} : d \rightarrow VC_{MN} : s$ から $P_{CN} : d \rightarrow G_{MN} : s$ に変換し，$P_{CN} : t \rightarrow G_{CM} : t$ の UDP カプセル処理を行って送信する．このパケットは NAT-A にて $G_{NATA} : u' \rightarrow G_{MN} : t$ に変換され，転送さ

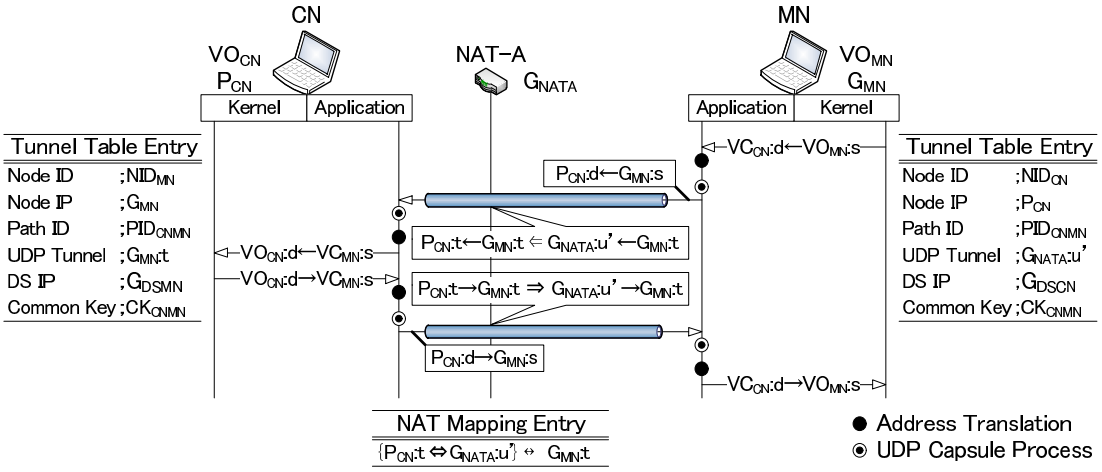

図 3.4　MN が CN との間で送受信するパケットの変換・カプセル処理

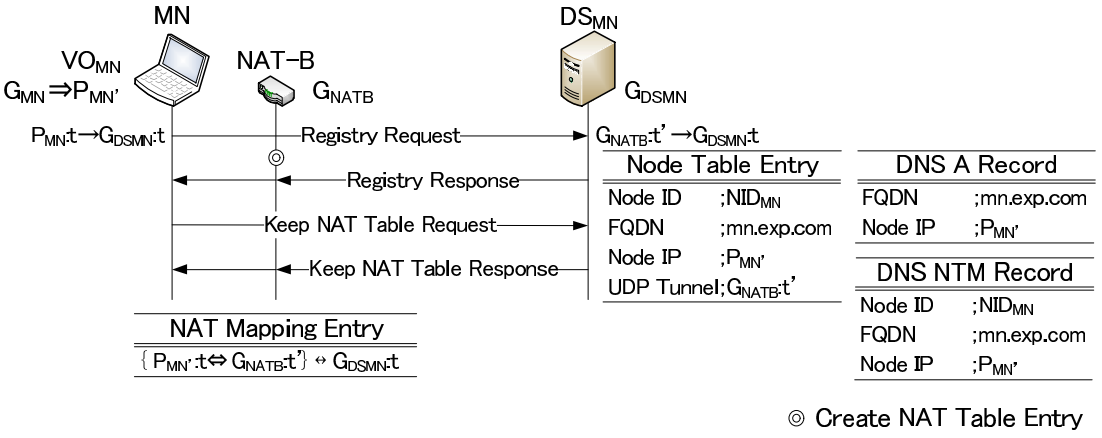

図 3.5　MN 移動後の登録処理

れる．MN はこれを受け取ると UDP カプセルを解除して $P_{CN} : d \rightarrow G_{MN} : s$ のアプリケーションパケットを取り出し，$VC_{MN} : d \rightarrow VO_{CN} : s$ に変換して上位アプリケーションに渡す．以上の処理により，アプリケーションパケットは NAT-A に影響されることなく CN と MN の間でやり取りされる．

3.3.2　MN が移動した時の処理

(1)　エンドノードの情報更新

図 3.2 の通信後，MN が IP アドレス G_{NATB} を持つ NAT-B 配下の，CN が存在するプライベートネットワークとは異なるプライベートネットワークへ移動して，図 3.1 (d) の構成で通信を行った場合の DS_{MN} への更新登録処理を図 3.5 に示す．

MN は移動後 $P_{MN'}$ を取得すると通信開始時と同様，DS_{MN} に $P_{MN'} : t \rightarrow G_{DSMN} : t$ の Registry Request を送信し，自身の情報（Node ID ”NID_{MN}”，物理アドレス ”P_{MN}”，FQDN

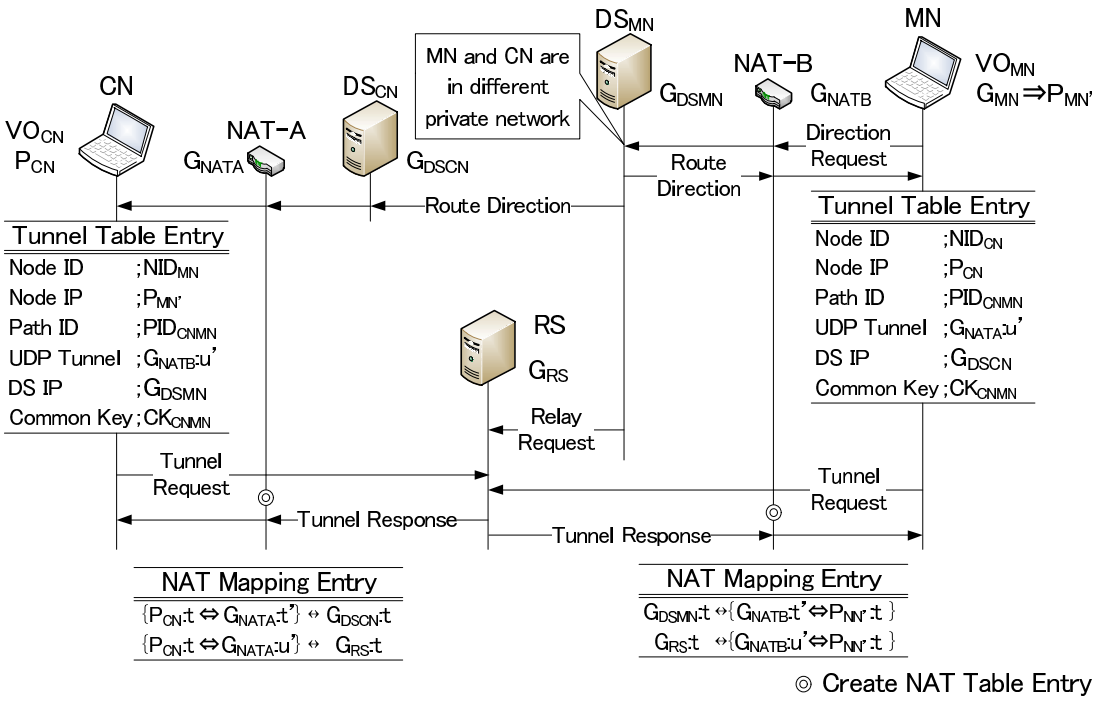

図 3.6　MN 移動後の UDP トンネル再生成ネゴシエーション処理

”*mn.exp.com*”）を通知する．途中，NAT-B はこのパケットにより，マッピングテーブルに以下のマッピングエントリを生成する．

$$\{P_{MN'} : t \Leftrightarrow G_{NATB} : t'\} \leftrightarrow G_{DSMN} : t$$

これにより，Registry Request は $G_{NATB} : t' \rightarrow G_{DSMN} : t$ に変換されて DS_{MN} に届けられ，DS_{MN} は記載された情報と IP ヘッダの送信元アドレス $G_{NATB} : t'$ から，MN に関する Node Table エントリの物理アドレスを $P_{MN'}$ に，NAT の IP アドレスを G_{NATB} に，UDP トンネル情報を $G_{NATB} : t'$ にそれぞれ更新する．また MN の A レコードの IP アドレスを G_{NATB} に，NTM レコードの MN の物理アドレスを $P_{MN'}$ に更新する．

(2)　UDP トンネル更新ネゴシエーション

MN 移動後の UDP トンネル再生成ネゴシエーション処理を図 3.6 に示す．本手法では移動前後に関わらず，DS からの指示に従って UDP トンネルを生成するといった UDP トンネル生成ネゴシエーション処理は全く同一であり，両エンドノードの通信構成により DS から受ける指示内容のみ異なる．この構成の場合，両エンドノードは RS との間でそれぞれ UDP トンネルを生成し，RS を介した中継通信を行なう．なお，RS は両エンドノードの中継通信に関する情報を管理する Relay Table を持つ．

DS_{MN} は Direction Request を MN から受けると，CN と MN が属する NAT-A，NAT-B の IP アドレス G_{NATA}，G_{NATB} が異なることから，両エンドノードがそれぞれ異なるプライベートネットワークに存在すると判断する．そして DS_{MN} は CN，MN に対し，G_{RS} に Tunnel

Request を送信するように指示する Route Direction を送信する．このとき，DS_{MN} は同時に CN と MN の通信を示す Path ID "PID_{CNMN}"と，その通信で使用する共通鍵 "CK_{CNMN}" を記載した Relay Direction を RS に送信し，RS に対して CN，MN からの Tunnel Request を受信するように指示する．

CN は Route Direction を受けると，MN に関する Tunnel Table エントリの物理アドレスを $P_{MN'}$ に，NAT の IP アドレスを G_{NATB} に，UDP トンネル情報を $G_{RS} : t$ に更新する．また VRT Table を以下のように更新する．

$$\{VO_{CN} \Leftrightarrow P_{CN}\} \leftrightarrow \{VC_{MN} \Leftrightarrow P_{MN'}\}$$

MN も同様，Route Direction を受けると，CN に関する Tunnel Table エントリの UDP トンネル情報を $G_{RS} : t$ に更新し，VRT Table を以下のように更新する．

$$\{P_{CN} \Leftrightarrow VC_{CN}\} \leftrightarrow \{P_{MN'} \Leftrightarrow VO_{MN}\}$$

また RS は Relay Direction を受けると，CN と MN の中継転送に使用する Relay Table エントリを生成し，Path ID "PID_{CNMN}"と，その通信で使用する共通鍵 "CK_{CNMN}"を登録する．

その後 CN，MN は指示に従い，それぞれ CN と MN の通信を示す Path ID "PID_{CNMN}" を記載した $P_{CN} : t \rightarrow G_{RS} : t$，$P_{MN'} : t \rightarrow G_{RS} : t$ の Tunnel Request を送信する．この時，NAT-A のマッピングテーブルには以下のマッピングエントリが生成される．

$$\{P_{\mathrm{CN}} : t \Leftrightarrow G_{\mathrm{NATA}} : u'\} \leftrightarrow G_{\mathrm{RS}} : t$$

同様に NAT-B のマッピングテーブルには以下のマッピングエントリが生成される．

$$G_{\mathrm{RS}} : t \leftrightarrow \{G_{\mathrm{NATB}} : u' \Leftrightarrow P_{\mathrm{MN'}} : t\}$$

以上の処理により，MN と CN の間に RS を介した UDP トンネルが生成され，以後は通信開始時と同様，この UDP トンネルの使用と VRT Table による変換を行って通信する．

(3) RS を介した中継通信処理

MN は CN の間の RS を介した UDP トンネルによる通信路の生成と互いの情報の更新が完了すると，アプリケーションパケットの送受信を再開する．MN が移動後に CN との間で送受信するパケットの変換・カプセル処理を図 3.7 に示す．

MN の上位アプリケーションは $VO_{MN} : s \rightarrow VC_{CN} : d$ のパケットを生成すると，カーネルモジュールにて VRT Table を参照し，アドレスを $P_{MN'} : s \rightarrow P_{CN} : d$ に変換した後，$P_{MN'} : t \rightarrow G_{RS} : t$ で UDP カプセル化して送信する．

送信されたパケットは NAT-B で $G_{NATB} : u' \rightarrow G_{RS} : t$ に変換され，RS に届けられる．RS はこのカプセルを解除して $P_{MN'} : s \rightarrow P_{CN} : d$ のアプリケーションパケットを取り出し，Relay Table から $G_{RS} : t \rightarrow G_{NATA} : u'$ で UDP カプセル化して送信する．

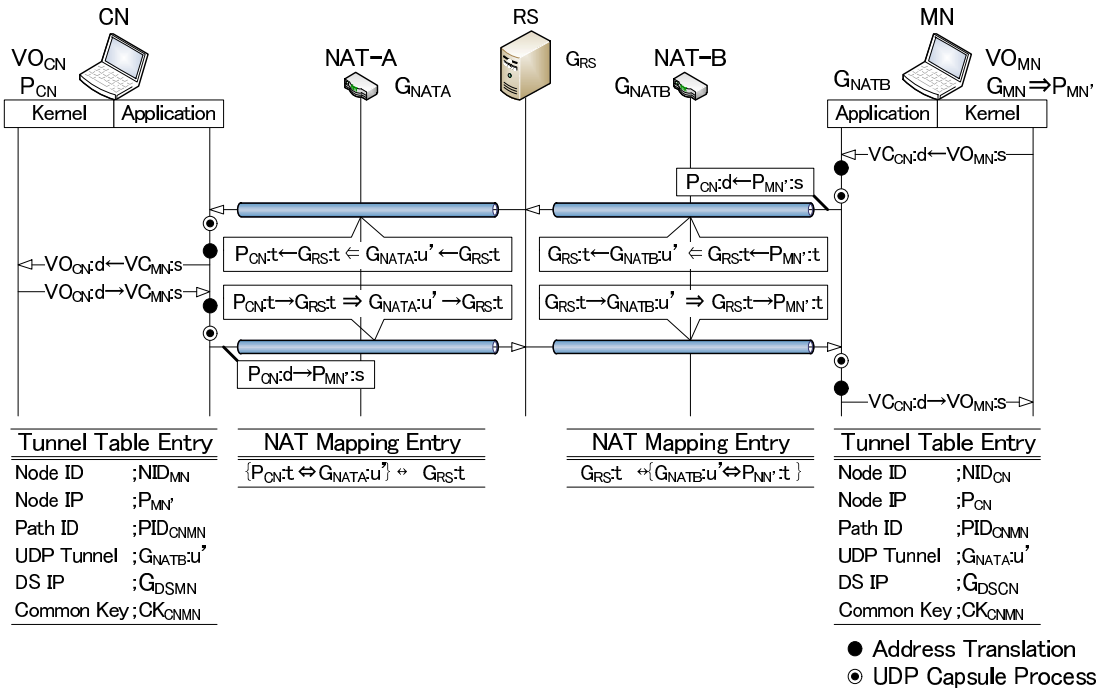

図 3.7　MN 移動後のパケットの変換・カプセル処理

このパケットは NAT-A に送り届けられ，NAT-A はマッピングテーブルを参照して IP アドレスを $G_{RS} : t \rightarrow P_{CN} : t$ に変換し，CN に転送する．CN はカーネルモジュールにて UDP カプセルを解除し，Tunnel Table を参照して $P_{MN'} : s \rightarrow P_{CN} : d$ を $VC_{MN} : s \rightarrow VO_{CN} : d$ に変換し，上位アプリケーションに渡す．CN が MN に送信する場合も同様の処理が行われる．以上の処理により，アプリケーションパケットは NAT-A，NAT-B に影響されることなく CN と MN の間でやりとりされ，また移動による IP アドレスの変化もアドレス変換によって吸収されるため，通信を継続したまま移動できる．

第4章　実装に向けた設計

NTMobile は移動通信を目的としているため，小型のノート PC やタブレット機，特に中でもスマートフォンを代表とした小型携帯端末での使用を想定している．現在，スマートフォンに搭載されている OS（ミドルウェア）を大別すると，iPhone や iPad に使用されている Apple 社の iOS と Google 社が開発している Android がある．

NTMobile では，エンドノードや RS が全てのアプリケーションパケットに対し，アドレス変換処理とカプセル／デカプセル処理を行う．これらの処理はアプリケーションのスループットに直結するため，カーネルで行うことが望ましい．そこで本提案では，オープンソースの OS である Android を実装対象とする．なお，Android ではカーネルに Linux を使用しており，Linux では作成したモジュールをカーネルから読み込むことができるため，全体のシステムへの影響を最小限に済ませることが可能である．

また，エンドノードと DS 及び RS 間，また DS と RS 間で行う UDP トンネルの生成ネゴシエーション処理はアプリケーションデーモンとして実装する．以下では，これらのノードのモジュール構成について説明する．

4.1　エンドノードのモジュール構成

4.1.1　カーネルモジュール

エンドノードのカーネルに実装するモジュール（以下，NTM カーネルモジュール）を図 4.1 に示す．エンドノードは仮想インタフェース（I/F）を持つ．仮想 I/F は自身を示す仮想アドレスを設定するために用いられ，実際にパケットが送受信されるわけではない．一般の上位アプリケーションから送信されるパケットは，IP 層で NTM カーネルモジュールによりフックされる．その後このモジュールはこのパケットに対して Tunnel Table を参照して仮想アドレスから物理アドレスへの変換，カプセル化，暗号化，MAC 値の設定を行ない，元の IP 層のモジュールに差し戻す．その後は通常の物理インタフェースから端末外部に送信される．

一般アプリケーションパケットを受信した際は，NTM カーネルモジュールはこれをフックし，DNS レスポンス以外のアプリケーションパケットであれば送信時と逆の処理を行った後，元の IP 層のモジュールに差し戻す．DNS レスポンスであれば一時退避し，Linux でユーザーランドとカーネル間でデータの受け渡しを行う汎用ソケットである Netlink ソケットを用い，NTM のアプリケーションデーモンに NTM レコードの DNS 問

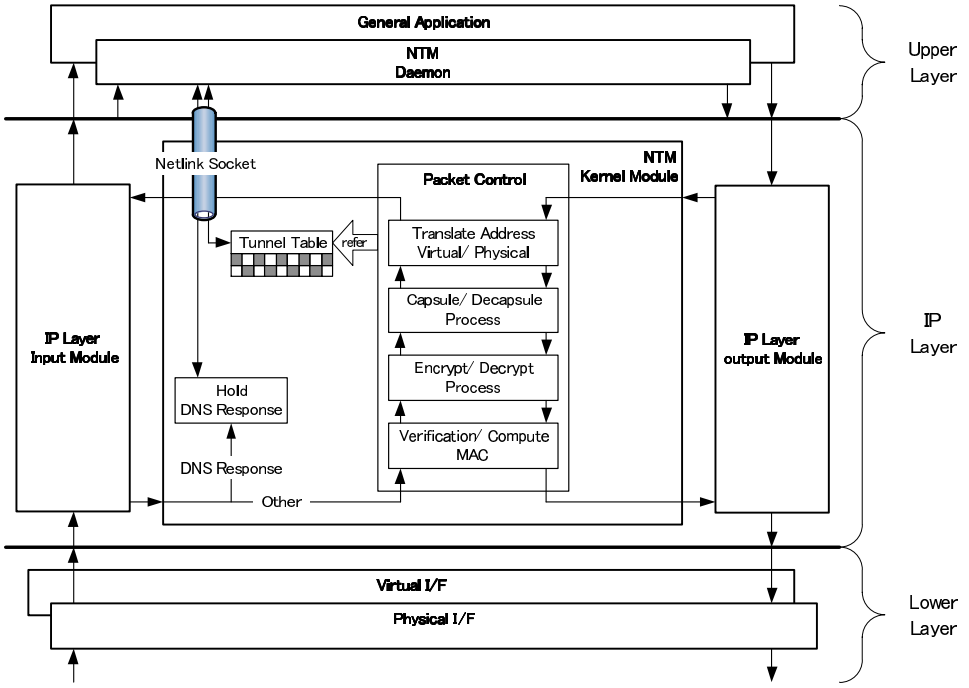

図 4.1　エンドノードの NTM カーネルモジュール

い合わせを行うよう通知する．また，NTM カーネルモジュールはアプリケーションに実装するデーモン（以下，NTM アプリケーションデーモン）から Netlink ソケットを用いて Tunnel Table のエントリ生成，削除，更新，及び退避した A レコードの DNS レスポンスの開放指示が通知される．なお，NTM の制御メッセージは NTM カーネルモジュールでは何も処理されない．Tunnel Table の生存時間は UNIX 時間により管理され，更新や変換等の参照が行われない場合は一定時間後に削除される．

4.1.2　アプリケーションデーモン

NTM アプリケーションデーモンはネゴシエーションデーモン，登録デーモン，移動通知デーモンの三つで構成される．エンドノードの NTM アプリケーションデーモンを図 4.2 に示す．

登録デーモンは DS へのエンドノードの情報登録とマッピングエントリの保持のため，Registry Request や Keep NAT Table Request の送信と，Registry Respons や Keep NAT Table Response の受信を行なう．ネゴシエーションデーモンは UDP トンネルの生成ネゴシエーションを行う際に使用する Route Direction や Tunnel Request，Tunnel Response の送信と，Route Direction や Tunnel Request，Tunnel Response を受信する．この時受け取ったデータは Netlink ソケットを使用し，Tunnel Table に書きこむ．NTM の DNS 問い合わせ処理を行うモジュールでは，Netlink ソケットを通じた NTM カーネルモジュールからの指示に従い，NTM レコードの DNS 問い合わせを行うため，このモジュールは NTM レコードの DNS Request の送信と DNS Response の受信を行い，これを完了すると，Netlink ソケッ

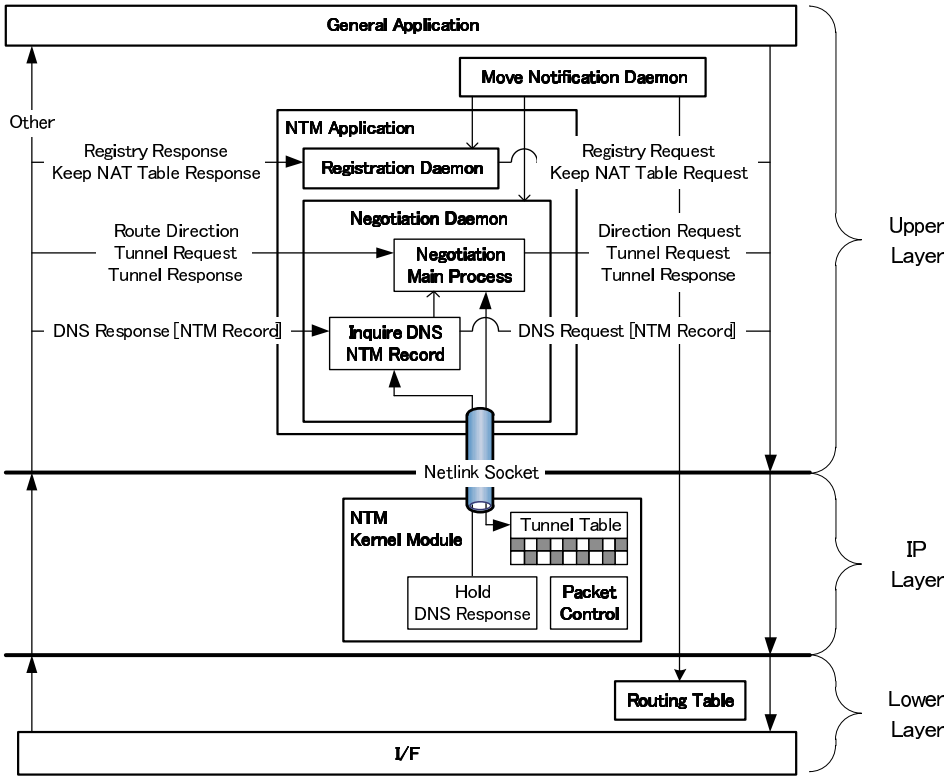

図 4.2　エンドノードの NTM アプリケーションデーモン

トを通じてカーネルモジュールに A レコードの DNS Response の開放を指示する．移動通知デーモンはルーティングテーブルを監視し，物理アドレスの値に変更があるとエンドノードが移動したと判断し，登録デーモンとネゴシエーションデーモンに移動の通知を行う．移動の通知を受けたそれぞれのデーモンは登録処理や UDP トンネルの生成ネゴシエーション処理を開始する．

4.2　DS のモジュール構成

4.2.1　アプリケーションデーモン

DS の NTM モジュールを図 4.3 に示す．DS で行われる処理は全てアプリケーションデーモンで行われる．

登録デーモンはエンドノードからの Registry Request を受信して受け取ったデータを Node Table に書き込む．更にそれらの情報を DNS の A レコードエントリや NTM レコードエントリに書き込む．これらの処理が終わると Registry Response を応答する．また Keep NAT Table Request を受信し，その応答として Keep NAT Table Response を送信する．ネゴシエーションデーモンは Route Direction を受信すると転送対象のエンドノードの Node Table エントリから UDP の通信路の情報を取得し，見つかった UDP の通信路を用いてエンドノードに転送する．Direction Request を受信した場合は記載されたデータ

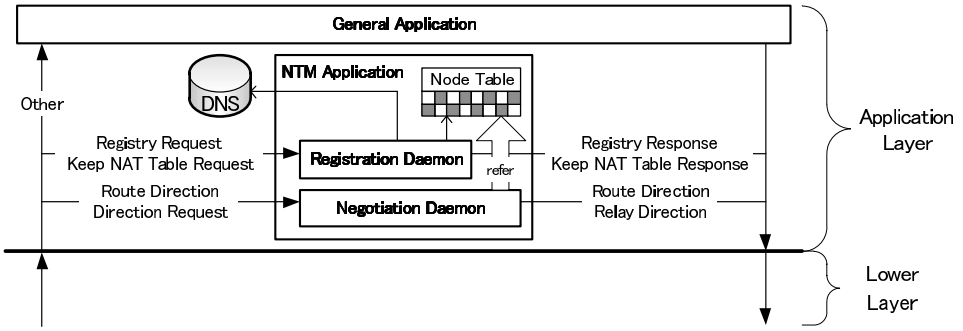

図 4.3　DS の NTM モジュール

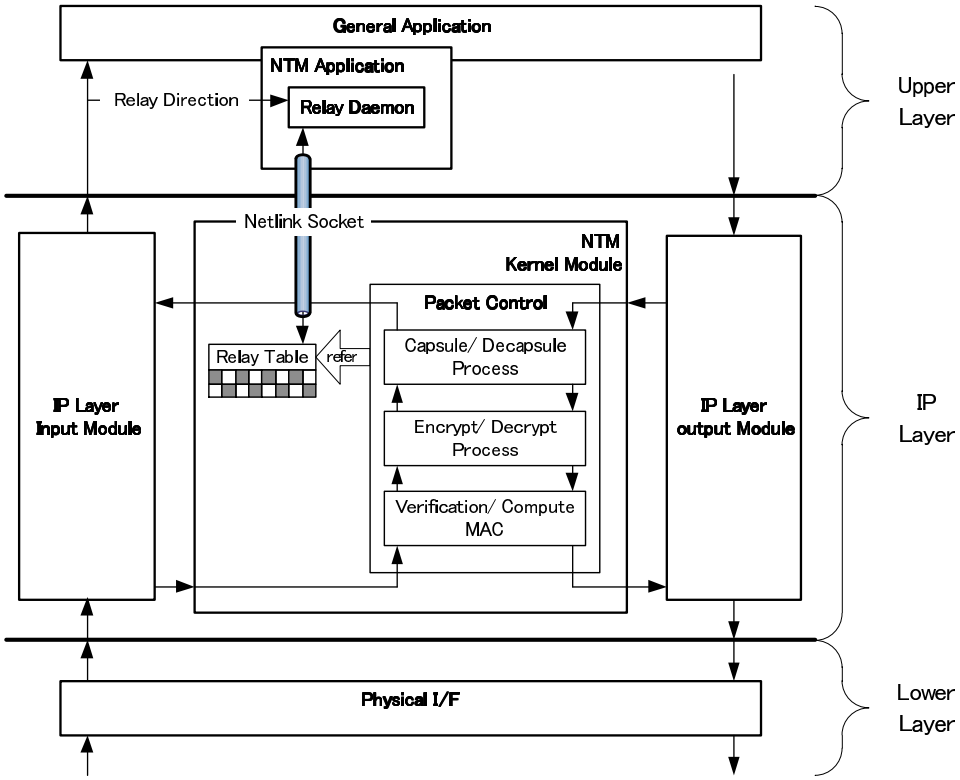

図 4.4　RS の NTM モジュール

から指示内容を決定し，Route Direction と，RS を中継した UDP トンネルの生成を指示する場合は Relay Direction も送信する．なお，Node Table のエントリの生存時間は Tunnel Table のエントリ同様，UNIX 時間により管理され，Registry Request による更新や Keep NAT Table Request による生存確認が行われない場合は，一定時間後に削除される．

4.3　RS のモジュール構成

RS の NTM モジュールを図 4.4 に示す．RS の NTM カーネルモジュールの構成は基本的にエンドノードの構成と同一である．ただし RS は仮想アドレスを持たないため，仮想

I/F も存在せず，仮想アドレスと物理アドレスの変換も行われない．また DNS に関する処理も存在しない．NTM アプリケーションデーモンでは Relay Direction のみ受信し，受け取ったデータを Netlink ソケットを介して NTM カーネルモジュールの Relay Table に書き込む．Relay Table のエントリの生存時間も Tunnel Table のエントリ同様，UNIX 時間により管理され，転送処理等のエントリへの参照が行われない場合，一定時間後に削除される．

第5章　まとめ

本論文では IPv4 において，移動透過性と NAT 越えを同時に実現し，エンドノードがあらゆる移動を可能とする方式を提案した．提案方式では DS を導入することにより，両エンドノード間で適切な UDP トンネルによる通信路を形成する．これにより，SPI フィルタリングやプロトコルの制限を回避することができる．また仮想アドレスと RS を導入することにより，異なるプライベートネットワーク間でのプライベートアドレスの重複を回避することができ，また確実な通信を実現する．RS をプロキシサーバとして使用することで，一般サーバとの通信中でも移動が可能になる．

また本論文ではこの提案の実装について設計を行った．またエンドノード及び DS の間の UDP トンネル生成シーケンスの実装まで完成している．今後はその他 RS の実装やデータパケットのアドレス変換処理及び UDP カプセル処理等の残実装を完成させ，検証や性能測定をしていきたい．

謝辞

本研究にあたり，多大なる御指導と御教授を賜りました，渡邊晃教授には心から感謝いたします．

本論文を作成するにあたり，快く査読を引き受けて下さり，熱心にご指摘を頂きました，柳田康幸教授に感謝の意を表します．

本論文を作成するにあたり，快く査読を引き受けて下さり，熱心にご指摘を頂きました，宇佐見庄五准教授に感謝の意を表します．

また，本研究を進めるにあたり，常日頃からの御意見ならびに御助言を受け賜りました，鈴木秀和助教に深謝いたします．

最後に，本研究を進めるにあたり，数々の有益な御助言や御討論を賜りました，渡邊研究室の諸氏に感謝します．

参考文献

[1] Deering, S. and Hinden, R.: Internet Protocol, Version 6 (IPv6) Specification, RFC 2460, IETF (1998).

[2] Turányi, Z., Valkó, A. and Campbell, A. T.: 4+4: an architecture for evolving the Internet address space back toward transparency, *SIGCOMM Comput. Commun. Rev.*, Vol. 33, pp. 43–54 (2003).

[3] Ng, T. S. E., Stoica, I. and Zhang, H.: A Waypoint Service Approach to Connect Heterogeneous Internet Address Spaces, Technical report, USENIX Annual Technical Conference 2001 (2001). Boston, MA,.

[4] UPnP Forum: *Internet Gateway Device(IGD) Standardized Device Control Protocol V 1.0*, http://www.upnp.org/ (2001).

[5] Cheshire, S., Krochmal, M. and Sekar, K.: NAT Port Mapping Protocol (NAT-PMP), Internet Draft draft-cheshire-nat-pmp-03 (2008).

[6] Borella, M., Lo, J., Grabelsky, D. and Montenegro, G.: Realm Specific IP: Framework, RFC 3102, IETF (2001).

[7] Borella, M., Grabelsky, D., Lo, J. and Taniguchi, K.: Realm Specific IP: Protocol Specification, RFC 3103, IETF (2001).

[8] Kondo, K.: Capsulated Network Address Translation with Sub-Address(C-NATS), Internet Draft draft-kuniaki-capsulated-nats-05, IETF (2003).

[9] Francis, P. and Gummadi, R.: IPNL: A NAT-extended internet architecture, *SIGCOMM Comput. Commun. Rev.*, Vol. 31, pp. 69–80 (2001).

[10] 鈴木秀和，宇佐見庄五，渡邊　晃：外部動的マッピングにより NAT 越えを実現する NAT-f の提案と実装，情報処理学会論文誌，Vol. 48, No. 12, pp. 3949–3961 (2007).

[11] Ford, B., Srisuresh, P. and Kegel, D.: Peer-to-Peer Communication Across Network Address Translators, Technical report, Proc. USENIX Annual Technical Conference (2005). pp.179?192.

[12] Huitema, C.: Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs), RFC 4380, IETF (2006).

[13] Mahy, R., Matthews, P. and Rosenberg, J.: Traversal Using Relays around NAT (TURN): Relay Extensions to Session Traversal Utilities for NAT (STUN), RFC 5766, IETF (2010).

[14] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. and Schooler, E.: SIP: Session Initiation Protocol, RFC 3261, IETF (2002).

[15] Rosenberg, J.: Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols, RFC 5245, IETF (2010).

[16] Perkins, C.: IP Mobility Support for IPv4, RFC 3344, IETF (2002).

[17] 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).

[18] Ferguson, P. and Senie, D.: Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing, RFC 2827, IETF (2000).

[19] Levkowetz, H. and Vaarala, S.: Mobile IP Traversal of Network Address Translation (NAT) Devices, RFC 3519 (2003).

[20] 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).

[21] 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， Vol. 2008, No. 44, pp. 69–74 (2008).

[22] 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).

[23] Vixie, P., Thomson, S., Rekhter, Y. and Bound, J.: Dynamic Updates in the Domain Name System (DNS UPDATE), RFC 2136, IETF (1997).

[24] Leach, P., Mealling, M. and Salz, R.: A Universally Unique IDentifier (UUID) URN Namespace, RFC 4122, IETF (2005).

研究業績

研究会・大会等

1. 水谷智大, 鈴木秀和, 渡邊　晃, “NAT を跨る移動透過性を実現する Mobile PPC の提案”, マルチメディア，分散，協調とモバイル（DICOMO2010）シンポジウム論文集, Vol.2010, No.1, pp.281-287, July 2010.

2. 水谷智大, 鈴木秀和, 渡邊　晃, “移動透過性を考慮した NAT 越え通信の提案”, 情報処理学会研究報告, 2009-MBL51, Vol.2009, Nov.2009.

3. 水谷智大, 鈴木秀和, 渡邊　晃, “NAT 越えと移動透過性を同時に実現する内部仮想アドレスの提案”, マルチメディア，分散，協調とモバイル（DICOMO2009）シンポジウム論文集, Vol.2009, No.1, pp.1566-1571, July 2009.

4. 水谷智大, 鈴木秀和, 渡邊　晃, “Mobile PPC における仮想インタフェースの検討”, 情報処理学会第 71 回全国大会講演論文集, Mar.2009.

5. 水谷智大, 鈴木秀和, 渡邊　晃, “Mobile PPC における仮想インタフェースの提案”, 平成 20 年度電気関係学会東海支部連合大会論文集, Sep.2008.