NTMobileにおける相互接続性の確立手法と実装

鈴木秀和[†1] 水谷智大[†1] 西尾拓也[†2]
内藤克浩[†2] 渡邊晃[†1]

IPv4/IPv6 ネットワークが混在した環境において，ネットワークのアドレス空間/アドレス体系に依存することなく，ノードの相互接続性の確立と移動透過性を実現する NTMobile を提案する．NTMobile ではノード間にトンネルを構築した上で仮想 IP アドレスを用いたコネクションを確立する．本論文では IPv4 ネットワークにおける相互接続性の確立手法と実装方法を示す．提案方式を Linux に実装することにより，NAT 配下のノードに対してコネクションを確立できることを確認した．

Design and Implementation of Establishment Method of Interoperability on NTMobile

Hidekazu SUZUKI,[†1] Tomohiro MIZUTANI,[†1]
Takuya NISHIO,[†2] Katsuhiro NAITO[†2]
and Akira WATANABE[†1]

In this paper, we propose a network architecture, called NTMobile that can provide interactive conectivity and mobility of nodes in IPv4/IPv6 coexistence environment. An NTMobile compatible node creates a tunnel for the correspondent node and establishes a connection with their virtual IP addresses through the tunnel. This paper describes the establishment of the interactive conectivity in IPv4 network and the implementation method. Implementation of the proposed method on Linux confirmed that the node can establish a connection to the correspondent node located behind the NAT router.

†1 名城大学大学院理工学研究科
Graduate School of Science and Technology, Meijo University
†2 三重大学大学院工学研究科
Graduate School of Engineering, Mie University

1. はじめに

近年，スマートフォンやタブレットなど高性能な携帯端末が急激に普及しつつある．これらの移動端末は無線 LAN だけでなく，3G や WiMAX，LTE（Long Term Evolution）などの無線ブロードバンドサービスといった複数の手段によりインターネットに接続することが可能である．そのため，利用者の位置や無線ネットワークの状況に応じて最適な通信品質を選択するために，通信メディアを切り替えて通信を行う場面が一般的になりつつある．このように異なる無線システムを切り替える動作を垂直ハンドオーバと呼ぶが，無線システムを切り替えると同時に移動端末が接続するネットワークも変化するため，移動端末の IP アドレスが変化してしまう．インターネットで使用されている TCP/IP は IP アドレスを用いて通信端末間のコネクションを管理しているため，ネットワークの移動が発生するとコネクションが切断されてしまう．この問題を解決する技術を移動透過性技術と呼び，多くの実現手法が提案されている[1)]．

一方，現在の IP ネットワークの状況に着目すると，IPv4 アドレスの枯渇がいよいよ目前に迫ってきており，IPv6 への移行が徐々に進みつつある．しかし，IPv6 は IPv4 との下位互換性がない独立したプロトコルとして定義されているため，現在の IPv4 ネットワークを即座に IPv6 ネットワークへ移行することができない．そのため当分の間，IPv4 ネットワークと IPv6 ネットワークが混在した環境が続くものと想定されている．また，IPv4 ネットワークではグローバル IP アドレスの数が十分にないため，NAT（Network Address Translation）によりプライベートネットワークを構築して運用が行われている．このような異なるアドレス空間/アドレス体系が混在する環境において移動透過性を実現するためには，通信開始時や移動時における端末間の接続性を確実に確立する必要がある．

本論文では，IPv4 ネットワークを対象とした移動透過性技術における端末間の相互接続性に焦点を当てて議論を進める．IPv4 を対象とした移動透過性技術には，Mobile IPv4[2)]，MATv4[3)]，Mobile PPCv4[4)] などがある．IPv4 ネットワークでは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またがって移動することが考えられ，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 が介在することになる．移動透過性技術では移動ノード（MN; Mobile Node）の IP アドレスの変化を管理する装置を用意し，MN はハンドオーバ時に IP アドレスの変化を管理装置に通知する必要がある．ここで，MN と管理装置の間に NAT が存在すると，通知する 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生じ

る．この問題を解決するために，Mobile IPv4 では移動通知を UDP によりカプセル化したり，NAT に独自の機能を追加する等の対策がある[5)–7)]．しかし，MN は管理装置（HA; Home Agent）を常に経由した冗長な通信となってしまったり，特殊な NAT ルータの配下でしか移動透過性を実現できないなどの課題がある．MATv4 は MN と通信相手ノード（CN; Correspondent Node）および管理装置の間の通信経路上に NAT が存在しないことを前提としている．これは NAT 配下のノードへの到達性を確保する手段を持ち合わせていないためであり，結果として NAT をまたがった移動はできない．

Mobile PPCv4 は著者らが提案している移動透過性技術であり，アドレスの変化を MN と CN 間で直接交換することにより，特別なアドレス管理装置を必要としない方式である．Mobile PPCv4 では Hole Punching を応用した手法や，NAT 越えを実現する NAT-f（NAT-free protocol）[8)] を組み合わせた方式を提案してきた[9),10)]．しかし，近年の NAT ルータに標準的に搭載され始めている SPI（Stateful Packet Inspection）と呼ぶフィルタリング技術により，移動後の TCP パケットが破棄されてしまったり，NAT-f を実装した特別な NAT ルータが設置されていなければ接続性を確保できないなどの課題がある．

これらの課題を解決するために，本論文では NAT に一切の機能を追加することなく，かつ移動先ネットワークを限定しない移動透過性技術として NTMobile（NAT Traversal with Mobility）を提案する．NTMobile では，エンドノードに仮想 IP アドレスを割り当てることにより，移動時の IP アドレスの変化を隠蔽し，IP 層より上位層ではアドレス空間に依存しない仮想的なコネクションを確立する．このコネクションを実ネットワーク上で確立するために，通信開始時にエンドノード間で UDP トンネルを構築する．通信開始ノードは DNS による名前解決時に通信相手ノードに接続するために必要なネットワーク位置情報を収集し，NAT の有無に応じて最適経路を実現できるトンネルを構築する．これにより，エンドノード間の通信経路上に NAT が存在しても，アドレス空間に影響されない相互接続を実現する．なお，ノードが移動した際には通信開始時と同じトンネル構築手順を行うことにより，トンネル経路の再構築と移動透過性を同時に実現することができる．そのため本論文では通信開始時と移動時に行うトンネル構築方法と実装について詳述する．

以下，2 章で提案方式の概要，3 章で相互接続性の確立手法，4 章で実装方法とプロトタイプシステムの動作結果を示す．5 章で関連技術を取り上げ，6 章でまとめる．

2. NTMobile の概要

NTMobile は次の要求を満たすことができるネットワークアーキテクチャである．

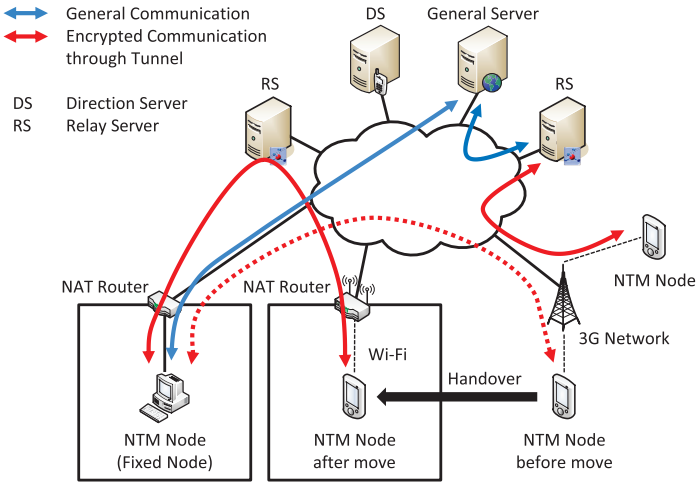


図 1 NTMobile の概要
Fig. 1 Overview of NTMobile system.

アドレス空間/アドレス体系に影響されない汎用性： NTMobile に対応したノード（NTM ノード）は IPv4/IPv6 に対応しており，グローバル IPv4 ネットワーク，NAT 配下のプライベート IPv4 ネットワーク，グローバル IPv6 ネットワークに接続できる．

相互接続性の実現： 通信相手 NTM ノードがプライベート IPv4 ネットワークに存在していても，通信を開始することができる．また IPv4/IPv6 の違いを意識することなく通信を開始することができる．

移動透過性の実現： NTM ノードは通信中に別の IPv4/IPv6 ネットワークにシームレスなハンドオーバができる．

図 1 に NTMobile の概要を示す．システムの構成要素として NTM ノードの他に，NTM ノードのネットワーク位置情報を管理する Direction Server（DS），異なる NAT 配下のプライベートネットワークに存在する NTM ノード間の通信を中継する Relay Server（RS）を定義する．NTM ノードはパケットロスのないシームレスハンドオーバを実現するために，無線 LAN や 3G，WiMAX など複数の無線通信技術を実装しているものとする．プライベートネットワークを構成する NAT は，SPI などのフィルタリング機能を実装している一般的な NAT ルータを想定しており，NTMobile に関わる特別な機能は一切持たない．

すべての NTM ノードはネットワーク接続時に DS に対して位置登録処理を行う．位置登録処理では，NTM ノードの実 IP アドレス，ノード ID，FQDN に加えて，NTM ノードがプライベートネットワークに存在する場合は NAT のグローバル IP アドレスが登録される．この時，NTM ノードは DS から仮想 IP アドレスが割り当てられ，NTM ノード間の通信に利用する．NTM ノードは通信開始時に相手 NTM ノードとの間に UDP トンネルを構築し，仮想 IP アドレスを用いてコネクションを確立する．UDP トンネルを用いることにより，NTM ノード間の通信経路上に SPI に対応した NAT が存在しても確実にコネクションの確立を実現することができる．また，仮想 IP アドレスを用いることにより，移動に伴う NTM ノードの実 IP アドレスの変化を隠蔽して移動透過性を実現する．さらに NTM ノード間の通信はエンドエンドで暗号化され，盗聴の防止や改ざんの検出が可能である．

NTMobile では，できる限りエンドツーエンド通信が行えるように，DS が通信ペアとなる NTM ノードのネットワーク位置情報に応じて，NTM ノードに最適なトンネル構築を指示する．この指示は通信開始時だけでなく，NTM ノードが様々なネットワークへ移動した時にも行われる．どちらか一方の NTM ノードがグローバルネットワークに存在すれば，他方は NAT 配下のプライベートネットワークにいても，RS を中継しない最適経路による通信を実現できる．なお，RS による中継通信が行われるのは，2 台の NTM ノードが異なるプライベートネットワークに存在する場合と，通信相手が NTMobile に対応しない一般ノードの場合だけである．

NTM ノードは移動を前提としているが，例えばデスクトップ PC のように移動することがないノードの場合は移動のサポートに係わる機能だけを除いたモードとしても動作できる．このような NTM ノードは NAT 配下のプライベートネットワークに存在していても，外部の NTM ノードからの接続性を確立することができる．また，一般ノードと通信する場合は仮想 IP アドレスを用いず，RS を中継しない通常のエンドツーエンド通信を行う．

DS は Dynamic DNS 機能を有しており，NTM ノードのアドレス管理や暗号鍵の生成，配布も行う．この他に，NTM ノードに割り当てる仮想 IP アドレスプールの保持や，NTM ノードに対してトンネル構築指示を行う役割を担っている．DS と RS はグローバルネットワーク上に設置し，ネットワークの規模に応じて分散設置が可能である．

なお，本論文の範囲外だが，NTMobile は IPv6 ネットワークへの対応も検討している．IPv4 ネットワークと IPv6 ネットワークの境界に RS を設置し，IPv4 と IPv6 の橋渡しを行う．NTM ノード間では仮想 IP アドレスによるコネクションを確立するが，接続しているネットワークのアドレス体系に応じて，上記 RS との間にトンネルを構築して転送する．これにより，IPv4 と IPv6 が混在した環境においても NTM ノードへの接続性および移動透過性を実現する．

3. 相互接続性の確立手法

本章では，NTMobile における NTM ノード間のコネクション確立手順について詳述する．NTMobile では通信を行うペアが本方式に対応していれば，双方とも移動が可能であるが，以後の説明では通信開始側 NTM ノードを MN，通信相手側ノードを CN として説明する．なお，用語の定義として，MN と CN を区別しない場合はエンドノードと表記し，本論文で用いる記号は付録 A.1 に示す．

3.1 前提条件

エンドノードはネットワーク接続時の位置登録処理[11] を完了しており，DS_N にはエンドノード N のネットワーク位置情報が登録されているものとする．エンドノードが使用する仮想 IP アドレスは DS により割当が行われ，重複がないものとする．なお，DS_N とエンドノード N 間，各 DS 間および各 DS と RS 間には信頼関係があるものと仮定する．

3.2 通信シーケンス

NTM ノードが通信を開始するまでの手順は**図 2** に示す名前解決，トンネル構築，暗号化通信の 3 つのフェーズで構成される．NTMobile システムでは**表 1** に示す通信パターンを想定しており，エンドノードが存在しているネットワークのアドレス空間の違いに応じて，最適な通信経路が確立できるようにトンネル確立フェーズのシーケンスが変化する．基本的な考え方として，通信ペアとなる NTM ノードのうち，どちらか一方がグローバルネット

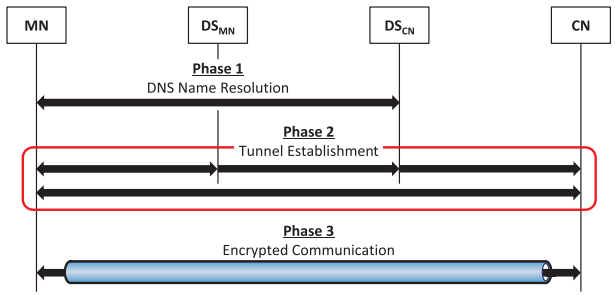


図 2 NTMobile シーケンス
Fig. 2 NTMobile sequence.

表 1 NTMobile で想定する通信パターンとトンネル経路
Table 1 Communication patterns and its tunnel route in NTMobile.

Pattern	Location of Initiator's NTM Node	Location of Correspondent Node	Tunnel Route
1	Global	Global (NTMobile)	End-to-End
2	Private	Global (NTMobile)	End-to-End
3	Global	Private (NTMobile)	End-to-End
4	Private1	Private2 (NTMobile)	via RS
5	Private2	Private2 (NTMobile)	End-to-End/via RS
6	Anywhere (Mobile Node)	Global (General)	via RS
7	Anywhere (Fixed Node)	Global (General)	End-to-End (No Tunnel)

ワークに接続している場合は，エンドエンドでトンネルを構築する．両ノードともプライベートネットワークに存在したり，通信相手が一般ノードの場合は RS を中継したトンネルを構築する．

ここでは，NTMobile 対応の MN・CN の一方または両方が NAT 配下のプライベートネットワークに存在している 3 つのパターン（Pattern 2〜4）を場合を取り上げて，通信シーケンスを示す．

3.2.1 名前解決フェーズ

MN は DNS により CN の名前解決を行い，DS_{CN} に登録されている RIP_{CN} が記載された DNS クエリの応答を受信する．ここで，MN はカーネルで DNS クエリ応答を一時待避させ，DS_{CN} へ NTMobile 専用レコードの問合せを行う．CN が NTM ノードであれば，MN は DS_{CN} から NTMobile 専用レコードを入手でき，CN に関する追加情報（NID_{CN}，RIP_{CN}，VIP_{CN}，$RIP_{NAT_{CN}}$ ⋆1，$RIP_{DS_{CN}}$）を取得，記録する．ここで，MN は 3.2.2 項に示すトンネル構築フェーズを実行し，完了後に待避していた DNS クエリ応答メッセージに記載されている RIP_{CN} を VIP_{CN} に書き換えてから，DNS リゾルバへ渡す．これにより，MN の上位アプリケーションは CN のアドレスを VIP_{CN} と認識することになる．

CN が NTMobile に対応していない一般ノードの場合，NTMobile 専用レコードの応答は得られないが，MN は RS に対してトンネル構築処理を行い，完了後に DNS クエリ応答を書き換えてから DNS リゾルバへ渡す．なお，MN が移動をサポートしない NTM ノードの場合は，DNS クエリ応答の書き換えを行わず，RIP_{CN} をそのまま DNS リゾルバへ渡す．

⋆1 CN がプライベートネットワークに存在している場合．グローバルネットワークに存在しているときは，RIP_{CN} となる．

3.2.2 トンネル構築フェーズ

MN は DS_{MN} へ Direction Request メッセージを送信する．このメッセージには MN 自身の情報（NID_{MN}，RIP_{MN}，VIP_{MN}）と NTMobile 専用レコードにより入手した CN の情報，および CN との間に構築するトンネルの識別子 PID_{MN-CN} が記載されている．

DS_{MN} は受信した MN と CN のノード ID および各種 IP アドレス情報から，

- MN と CN はグローバルネットワークに存在する．
- MN はプライベートネットワーク，CN はグローバルネットワークに存在する．
- MN はグローバルネットワーク，CN はプライベートネットワークに存在する．
- MN と CN は異なるプライベートネットワークに存在する．
- MN と CN は同じプライベートネットワークに存在する．
- CN は NTMobile 非対応の一般ノードで，グローバルネットワークに存在する．

ことを判定し，トンネル構築手順を決定する．構築手順が決定した後，Route Direction メッセージにより各ノードにその後のトンネル構築動作を指示する．なお，Route Direction には DS_{MN} が生成した共通鍵 K_{MN-CN} を配布する役割も担う．

図 3 に 3 つの通信パターンを例にトンネル構築手順を示す．これらのパターンでは NAT の配下に存在するエンドノードに対して Route Direction を送信する必要がある．エンドノード N はネットワーク接続時の位置登録処理を定期的に DS_N に対して実行しているため，NAT_N にはエンドノード N 宛の制御メッセージを受け入れるためのポートが常に開けられている．そのため，DS_N は NAT_N に開けられているポート番号に向けて Route Direction を送信することにより，エンドノード N に対して動作指示を行うことができる．エンドノードに対する指示は下記の通りである．

Private-to-Global (Pattern 2)： MN と CN 間のエンドエンドでトンネルを構築する．このとき，MN はプライベートネットワークに存在するため，以後のシーケンスは MN 側から開始する必要がある．そのため，DS_{MN} は Route Direction により，MN には CN へ Tunnel Request メッセージを送信するよう指示する．一方，CN には DS_{CN} を経由して，MN からの Tunnel Request を受信するよう指示する．

Global-to-Private (Pattern 3)： Pattern 2 と同様に MN と CN 間のエンドエンドでトンネルを構築する．このとき，CN はプライベートネットワークに存在するため，以後のシーケンスは CN 側から開始する必要がある．そのため，DS_{MN} は Route Direction により，MN には CN からの Tunnel Request を受信するよう指示する．一方，CN には DS_{CN} を経由して，MN へ Tunnel Request を送信するよう指示する．

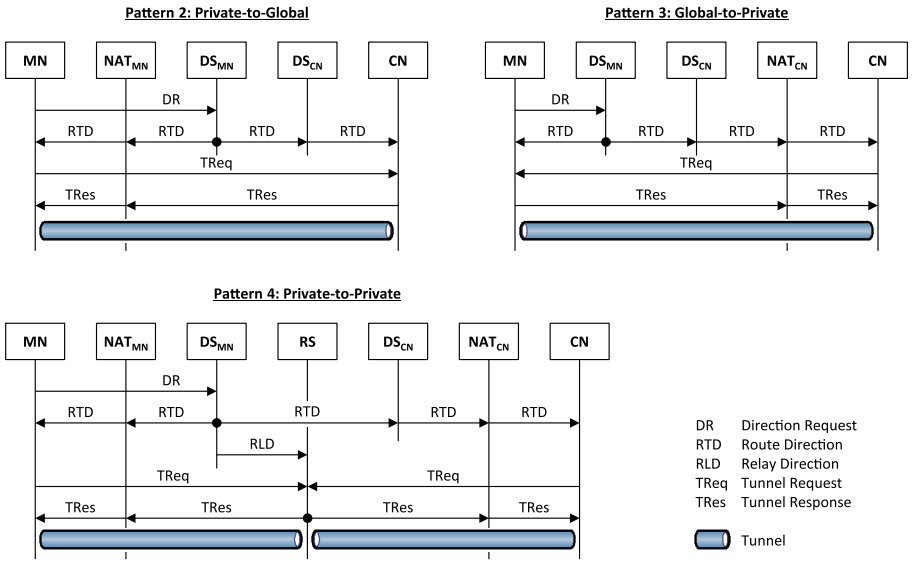


図 3 トンネル構築手順
Fig. 3 Tunnel establishment procedure.

Private-to-Private (Pattern 4)： MN と CN が別々のプライベートネットワークに存在するため，Relay Server との間にトンネルを構築する．このとき，以後のシーケンスは MN，CN 両側から開始する必要がある．そのため，DS_{MN} は Route Direction により，MN と CN には RS へ Tunnel Request を送信するよう指示する．さらに，Relay Direction により，RS には MN と CN からの Tunnel Request を受信するよう指示すると共に，共通鍵 K_{MN-CN} を配布する．以上の処理により，MN と CN は RS との間で共通鍵を共有することができる．

以後，DS からの指示に応じて該当ノード間で Tunnel Request/Response の交換を行い，トンネルテーブルを作成する．Tunnel Response を受信した MN はトンネル構築フェーズを終了し，待避していた DNS クエリ応答に対して 3.2.1 項で述べた書き換え処理などを行う．

3.2.3 暗号化通信フェーズ

MN は宛先が仮想 IP アドレスのパケットを送信する際，IP 層に生成されたトンネルテーブルに従って，元の IP パケットを UDP でカプセル化し，暗号化処理の後にトンネル構築対

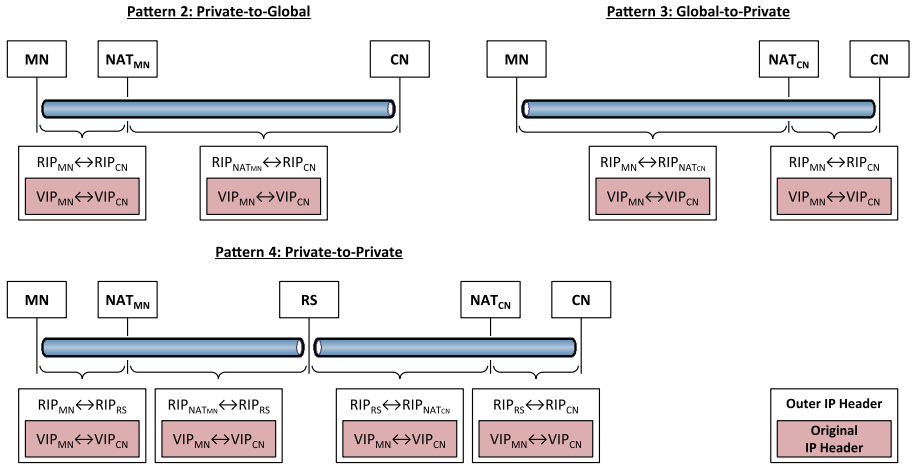


図 4 IP パケットのアドレス遷移
Fig. 4 Address trantisions of IP packet.

象ノードへ送信する．このとき，**図 4** に示すように，元の IP ヘッダには送信元を VIP_{MN}，宛先を VIP_{CN} としたままで，新たに付け加えられる IP ヘッダにはトンネルの両端の実 IP アドレスとなる．従って，エンドノードの通信経路上に NAT が存在する場合は，外側の IP ヘッダおよび UDP ヘッダが NAT によりアドレス変換されるため，カプセル化されたオリジナルの IP パケットは仮想 IP アドレスのまま維持される．トンネルの出口に当たるノードは，受信したパケットを復号，デカプセル化してから上位アプリケーションへ渡す．

以上の処理により，エンドノードの位置に応じた最適なトンネル経路が構築され，仮想 IP アドレスによる相互接続性を確立することができる．なお，同一エンドノード間であれば，構築された 1 つのトンネルで複数のコネクションをまとめて転送することができる．

3.3 移動時の対応

エンドノードが移動して実 IP アドレスが変化した場合，通信開始時とまったく同じトンネル構築フェーズを実行してトンネルの再構築を行う．これは移動先ネットワークのアドレス空間に応じて，エンドエンドまたは RS 経由のトンネルに切り替える必要があるためである．トンネルの再構築処理が完了しても，エンドノードの上位アプリケーションは常に仮想 IP アドレスに基づいたコネクションを確立しているため，実 IP アドレスの変化に影響されることはなく，移動透過性を実現できる[12]．

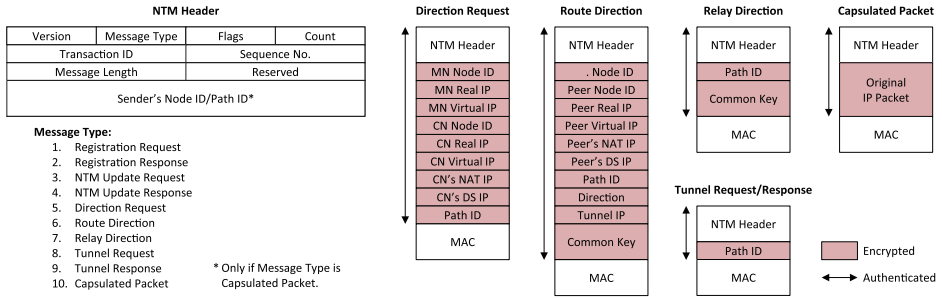


図 5 メッセージフォーマット
Fig. 5 Message formats.

3.4 メッセージフォーマット

図 5 に NTMobile におけるトンネル構築に関わる制御メッセージフォーマットを示す．制御メッセージは UDP プロトコルを利用し，NTM ヘッダと各制御メッセージペイロードで構成される．NTM ヘッダは制御メッセージの種類を判別する Type フィールドや，制御メッセージの送信ノードを示すノード ID などが記載される．DS と各ノードは信頼関係があることを前提としているため，Direction Request，Route Direction，Relay Direction はすべて事前に共有済みの暗号鍵を用いて暗号化および MAC（Message Authentication Code）の付与が行われる．Tunnel Request/Response およびトンネル通信の暗号化と認証には，トンネル構築フェーズで配布された共通鍵を利用する．

各ノードは NTM ヘッダに記載されている送信ノード ID をキーにして，復号や MAC の検証に用いる暗号鍵を決定する．カプセル化パケットではノード ID の代わりに，構築されたトンネル経路を示す Path ID が NTM ヘッダに記載されており，トンネルテーブルの検索時のキーとして利用する．

4. 実装と評価

NTMobile は Android OS を搭載した携帯端末での利用を想定しているため，本論文では Linux での実装方法について述べる[⋆1]．

⋆1 Android とは，米 Google 社がモバイル向けプラットフォームとして発表したオープンソースの OS である．カーネルとして Linux を採用しているため，今回実装したモジュールを Android へ移植することが可能である．

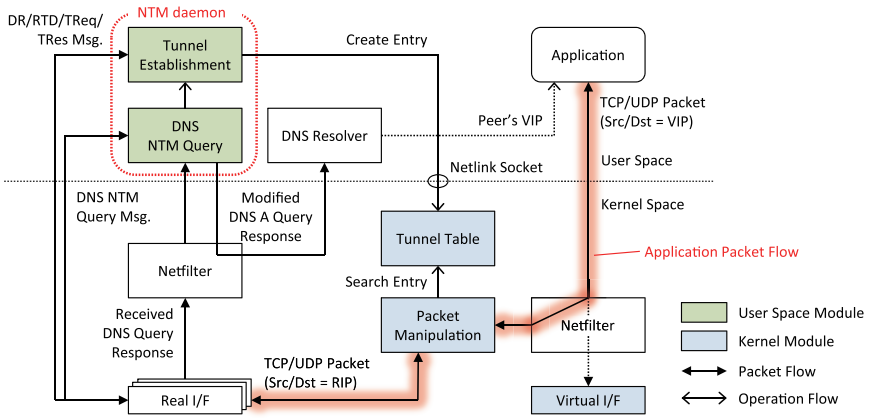


図 6 モジュール構成（NTM ノード側）
Fig. 6 Module configuration (NTM node).

4.1 NTM ノード

図 6 に NTM ノードのモジュール構成を示す．NTM ノード側にはユーザスペースで動作する NTM デーモンと，カーネルで動作するパケット操作モジュール，トンネルテーブルおよび仮想インタフェースを実装する．

NTM デーモンは Netfilter[⋆2]を利用して DNS クエリの応答をフックする．クエリ応答を解析して A レコードの結果を取得していたら，そのレコードを保持している DS へ NTMobile 専用レコードの問合せを行う．その後，トンネル構築フェーズの終了時に，Netlink ソケットを通じてカーネルに実装されているトンネルテーブルにエントリを登録する．

通常のアプリケーションが送信する IP パケットの宛先は仮想 IP アドレスとなっているため，仮想インタフェースに向けてカーネルへ渡される．仮想インタフェースに渡される IP パケットは Netfilter によりパケット操作モジュールに渡され，カプセル化，暗号化などの処理が行われ後，実インタフェースから送信される．受信時は逆の手順により復号，デカプセル化された後，アプリケーションへデータが渡される．一般にカプセル化を行うための仮想インタフェースとして，TUN/TAP デバイス[⋆3]がある．例えば OpenVPN では，

⋆2 http://www.netfilter.org/index.html
⋆3 http://vtun.sourceforge.net/tun/

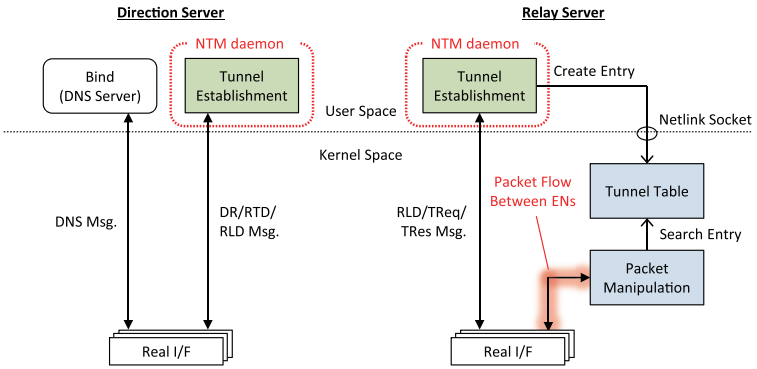
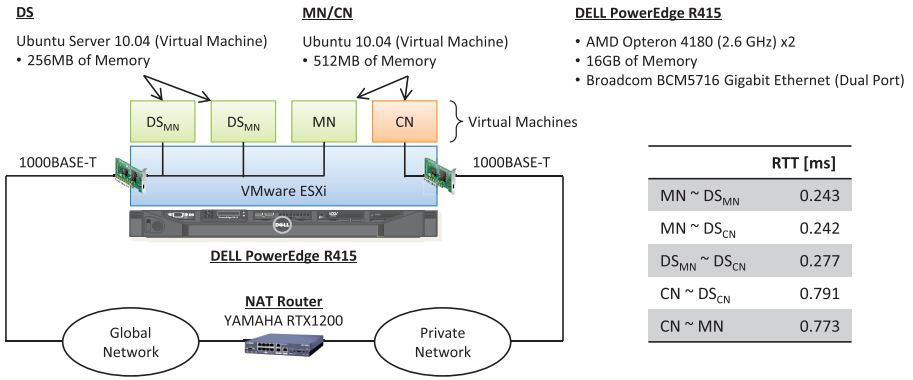


図 7 モジュール構成（左：DS 側，右：RS 側）
Fig. 7 Module configuration (Left: Direction Server, Right: Relay Server).



	RTT [ms]
MN ~ DS_{MN}	0.243
MN ~ DS_{CN}	0.242
DS_{MN} ~ DS_{CN}	0.277
CN ~ DS_{CN}	0.791
CN ~ MN	0.773

図 8 試験ネットワーク構成
Fig. 8 Test network configuration.

TUN/TAP デバイスに渡されたパケットデータを一度ユーザ空間へコピーしてからカプセル化を行うため，スループットが低下するという課題がある．これに対して，NTMobile ではパケットのカプセル化処理をすべてカーネル内で完結するように設計しており，冗長のないパケットフローと高スループットを実現する．

4.2 Direction Server と Relay Server

図 7 に DS と RS のモジュール構成を示す．DS と RS には，上述した NTM ノードのモジュールの一部を実装する．DS は NTM ノードのアドレス情報を動的に登録するために，Dynamic DNS サーバを稼働させる．この DNS サーバには NTMobile 専用レコードを扱えるよう，`bind`[*1]に機能を追加して実現している．NTM ノードおよび RS との制御メッセージ交換は NTM デーモンが行う．

RS は DS および NTM ノードとの制御メッセージ交換を行う NTM デーモンに加えて，カーネルモジュールとしてトンネルテーブルとパケット操作モジュールを実装する．なお，RS は受信したカプセル化パケットの転送処理が主な役割であるため，仮想インタフェースは実装しなくてもよい．

4.3 動作検証

提案方式による相互接続性の確立を検証するために，プロトタイプシステムを実装した．**図 8** に試験ネットワーク構成と各装置の仕様を示す．1 台の実機サーバにインストールした VMware ESXi4.1 を利用して，NTM ノードおよび DS を仮想マシンとして構築した．今回は表 1 の Pattern 3 の動作検証と通信開始時に発生するトンネル構築時間を測定するために，MN と 2 台の DS にグローバル IP アドレスを，CN にプライベート IP アドレスを割り当て，CN を NAT ルータの LAN 側に，その他を NAT ルータの WAN 側に接続した．各装置の仕様および平均 RTT は図中の通りである．このような構成において MN から CN へ ping を実行し，このときのパケットフローを Wireshark を用いて MN 側で観測した．なお，制御メッセージの暗号化および認証アルゴリズムは AES-CFB，HMAC-MD5 とし，各装置には制御メッセージを暗号化するための共通鍵（鍵長 128bit）を事前に設定した．

図 9 に通信開始時におけるトンネル構築時間を示す．DNS クエリ応答を受信してから最初の ICMP パケットを送信するまでに要した時間は，一般ノードの場合は 0.183 ms，NTM ノードの場合は 19.258 ms であった．トンネル構築時間の内訳に着目すると，名前解決フェーズの NTMobile 専用レコードの問合せが 0.519 ms であり，これは NTM ノードに実装したプログラムの処理負荷に当たる．NTMobile 専用レコードの応答受信からトンネル構築フェーズの Direction Request 送信までに 5.070 ms を要しており，これは DNS 応答メッセージの解析，制御メッセージの生成，暗号化および MAC 生成処理が含まれる．Pattern 3 では MN が Direction Request を送信すると，CN 側から送信される Tunnel Request を受信する．この時間が 8.767 ms であったが，この間に Route Direction が DS_{MN} から DS_{CN}，CN の順に転送されており，各装置でメッセージの復号・検証および暗号化・MAC

*1 `http://www.isc.org/software/bind`

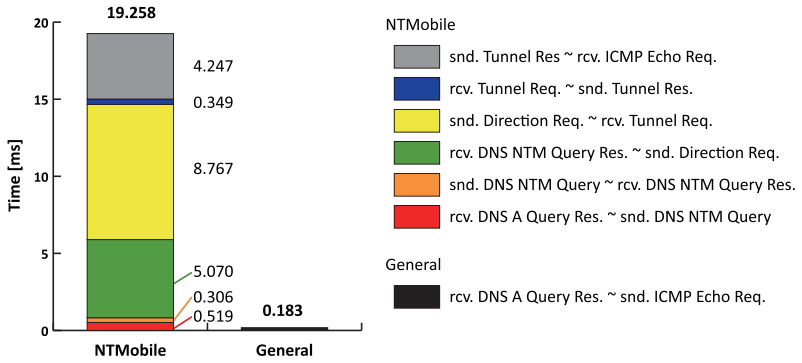


図 9 Pattern 3 における通信開始時のトンネル構築時間の結果
Fig. 9 The Result of initial time caused by the tunnel establishment in Pattern 3.

生成などの処理が行われている．MN が Tunnel Response を返答してから ICMP パケットを送信するまでに 4.247 ms かかっており，トンネルテーブルの作成，DNS 応答メッセージ内の IP アドレスを仮想 IP アドレスに書き換える処理などを行っている．

今回は仮想マシンを用いた環境での測定結果であるため，実環境では各装置間のエンドツーエンド遅延が上記結果に加わることになる．ここで，MN と DS_{MN} の位置を日本，CN と DS_{CN} の位置を米国と仮定すると，DS 間および MN と CN 間のエンドツーエンド遅延が大きくなる．Pattern 3 の場合，トンネル構築フェーズで日米間を 1.5 往復するため，約 150 ms のエンドツーエンド遅延が発生すると推測される．

例えば，NAT 越えを実現するための技術として ICE（Interactive Connectivity Establishment）[13] がある．ICE は SIP（Session Initiation Protocol）で NAT 越えをするための技術で，通信開始時に STUN（Session Traversal Utilities for NAT）[14] や TURN（Traversal Using Relays around NAT）[15] を反復的に実行し，エンドノードが互いに接続可能な IP アドレスとポート番号を発見・交換する．文献 16) によると，ICE によるコネクション確立に約 2〜10 秒必要であることが示されている．この結果と比較すると，提案方式のトンネル構築時間は十分短く，実用上問題ないと考えられる．

5. 関 連 技 術

異なるアドレス空間をシームレスに接続するアーキテクチャとして，SIPS（Seamless IP Sublayer）が提案されている[17]．SIPS は IP 層とトランスポート層の間に IP 層の副層を定義して既存の IP 層を拡張している．この副層においてラベルスイッチング技術により，ヘッダ変換，ホストの特定，パケットのルーティングなどを行うことにより，異なるアドレス空間での通信を実現している．SIPS ではエンドノードの IP 層拡張の他，各アドレス空間の境界に SIPS 対応ルータを配置する必要があるため，例えば IPv4 ネットワークでは NAT ルータの位置に該当する．そのため，本論文で議論している異なるアドレス空間に存在するノード間の相互接続性の実現には適用できるものの，SIPS 対応ルータが設置されていないネットワーク，すなわち一般の NAT ルータ配下に移動した場合は相互接続することができない．また，移動透過性については考慮されていない．

IETF では Mobile IP の他に HIP（Host Identity Protocol）と呼ぶ次世代モバイルプロトコルが標準化されている[18]．HIP は IP 層とトランスポート層に Host Identity 層を追加し，ID と Locator を分離して使い分ける．上位層ではノード識別子 HI（Host Identifier），下位層では IP アドレスを使用することにより，上位層に IP アドレスの変化を隠蔽して移動透過性を実現している．HIP は IPv4 ネットワークと IPv6 ネットワークをまたがった移動に対応しており，IPv4 における NAT 越えを実現するために ICE を用いた拡張仕様が定義されている[19]．そのため，ハンドオーバ時に ICE に基づくシグナリングを行うことを考慮すると，オーバヘッドが高いと想定される．

これまでに挙げた移動透過性技術はネットワーク層で実現する技術であったが，アプリケーション層で実現する技術もある．代表的な技術として SIP を用いた方式が多く，例えば All-SIP mobility では従来の UDP セッションだけでなく TCP コネクションの維持も可能な技術である[20),21)]．この技術はエンドノードが仮想 IP アドレスと UDP トンネルを用いて移動透過性を実現している点が提案方式と類似している．UDP カプセル化処理はアプリケーション層に実装された SMC（Session and Mobility Controller）モジュールが行うため，仮想インタフェースに渡された IP パケットを一度アプリケーション層へ戻す必要がある．一方，提案方式はカプセル化処理はカーネルモジュールで行うため，All-SIP mobility に対してスループット性能の優位性がある．

提案技術と類似したアプローチにより NAT をまたがった移動透過性を実現する技術として，UPMT（Universal Per-Application Mobility management using Tunnels）が提案されている[22),23)]．UPMT はアプリケーション層の移動透過技術で，NAT 配下に存在するノードはインターネット上に設置した AN（Anchor Node）に対してトンネルを構築し，AN は受信パケットをデカプセル化した後，NAT によるアドレス変換を行って通信相手ノードへ転送する．しかし，通信相手ノードがグローバルネットワークに存在していても常に AN

を中継したデータ転送となるため，提案方式と比較すると経路が冗長であること，また AN に負荷が集中するなどの課題がある．なお，UPMT においても ICE を用いてエンドツーエンド通信を行う拡張仕様があるが，前述の通りシグナリングのオーバヘッドが高いと考えられる．

6. ま と め

本論文では，移動先ネットワークの制約がない移動透過性技術として NTMobile を提案した．提案方式は DS を導入することにより，エンドノードのネットワーク位置に応じた最適なトンネル通信経路を確立できることを示した．仮想 IP アドレスによる実 IP アドレスの変化を隠蔽しているため，移動後にトンネルの再構築をおこなうことにより，NAT をまたがった移動透過性を実現できる．提案方式のプロトタイプシステムを Linux に実装したところ，NAT 配下のノードに対してコネクションを確立できることを確認した．

今後は実環境におけるシグナリングのオーバヘッドを評価し，ハンドオーバ時におけるトンネル再構築処理の影響を明らかにする．また，IPv4/IPv6 混在ネットワークにおける相互接続性の確立手順，各装置間の鍵管理などの仕様について検討する予定である．

謝辞 カーネルモジュールの実装に御協力頂いた東京システムハウス株式会社の関係各位に深謝する．

参 考 文 献

1) Le, D., Fu, X. and Hogrefe, D.: A Review of Mobility Support Paradigms for the Internet, *IEEE Communications Surveys & Tutorials*, Vol.8, No.1, pp.38–51 (2006).
2) Perkins, C.: IP Mobility Support for IPv4, Revised, RFC 5944, IETF (2010).
3) 関　顕生，岩田裕貴，森廣勇人，前田香織，近堂　徹，岸場清悟，西村浩二，相原玲二：IPv4 拡張した移動透過通信アーキテクチャMAT の設計と性能評価，情報処理学会論文誌，Vol.52, No.3, pp.1323–1333 (2011).
4) 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).
5) Levkowetz, H. and Vaarala, S.: Mobile IP Traversal of Network Address Translation (NAT) Devices, RFC 3519, IETF (2003).
6) Montenegro, G.: Reverse Tunneling for Mobile IP, revised, RFC 3024, IETF (2001).
7) 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).
8) 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 越え通信を実現する 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).
9) Suzuki, H. and Watanabe, A.: Design of NAT Traversal for Mobile PPC Applying Hole Punching Technology, *Proc. of IEEE TENCON2008* (2008).
10) 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).
11) 西尾拓也，内藤克浩，水谷智大，鈴木秀和，渡邊　晃，森香津夫，小林英雄：NTMobile における端末アドレスの移動管理と実装，DICOMO2011 論文集 (2011).
12) 内藤克浩，西尾拓也，水谷智大，鈴木秀和，渡邊　晃，森香津夫，小林英雄：NTMobile における移動透過性の実現と実装，DICOMO2011 論文集 (2011).
13) Rosenberg, J.: Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols, RFC 5245, IETF (2010).
14) Rosenberg, J., Mahy, R., Matthews, P. and Wing, D.: Session Traversal Utilities for NAT (STUN), RFC 5389, IETF (2008).
15) 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).
16) Maenpaa, J., Andersson, V., Camarillo, G. and Keranen, A.: Impact of Network Address Translator Traversal on Delays in Peer-to-Peer Session Initiation Protocol, *Proc. of IEEE GLOBECOM2010* (2010).
17) 角野宏光，内田良隆，石川憲洋，峰野博史，水野忠則：異なるアドレス空間をシームレスに接続する IP 層拡張の提案と実装，電子情報通信学会論文誌 (B)，Vol.J93-B, No.10, pp.1397–1407 (2010).
18) Moskowitz, R. and Nikander, P.: Host Identity Protocol (HIP) Architecture, RFC 4423, IETF (2006).
19) Komu, M., Henderson, T., Tschofenig, H., Melen, J. and Keranen, A.: Basic Host Identity Protocol (HIP) Extensions for Traversal of Network Address Translators, RFC 5770, IETF (2010).
20) Seta, N., Miyajima, H., Zhang, L., Hayashi, H. and Fujii, T.: All-SIP Mobility: Session Continuity on Handover in Heterogeneous Access Environment, *Proc. of IEEE VTC2007-Spring*, pp.1121–1126 (2007).
21) Miyajima, H., Zhang, L., Hayashi, H. and Fujii, T.: An Implementation of Enhanced All-SIP Mobility, *Proc. of IEEE PIMRC2008* (2008).
22) Bonola, M., Salsano, S. and Polidoro, A.: UPMT: Universal Per-Application Mobility Management Using Tunnels, *Proc. of IEEE GLOBECOM2009* (2009).
23) Bonola, M. and Salsano, S.: S-UPMT: a secure Vertical Handover solution based on IP in UDP tunneling and IPsec, *GTTI Riunione Annuale 2010*, (online), `http://www.gtti.it/GTTI10/papers/gtti10_submission_29.pdf` (2010).

付　　録

A.1　記号の定義

- RIP_N; ノード N の実 IP アドレス
- VIP_N; ノード N の仮想 IP アドレス
- NID_N; ノード N の識別子
- PID_{N1-N2}; ノード $N1$ とノード $N2$ 間で構築するトンネルの識別子
- K_{N1-N2}; ノード $N1$ とノード $N2$ の共通鍵
- $IP_{N1} \leftrightarrow IP_{N2}$; ノード $N1$ の IP アドレスとノード $N2$ の IP アドレス間のコネクション