Vol.2015-DPS-163 No.3
Vol.2015-MBL-75 No.3
2015/5/28

非root Android端末における IPv4/IPv6間シームレス接続性の確保

山田 貴之[1]　鈴木 秀和[1]　内藤 克浩[2]　渡邊 晃[1]

概要：今日のインターネットは，NAT（Network Address Translation）によるプライベートネットワークの構築や互換性のない IPv4 と IPv6 のネットワークが混在するなど，複雑な環境となっている．筆者らは，複雑化したインターネット環境において，モバイル端末同士が実ネットワークに依存することなくシームレスな通信を実現する仕組みとして，NTMobile（Network Traversal with Mobility）を提案している．これまで NTMobile では，一般のモバイル端末向け実装モデルとして，Android 4.0 以降で利用可能な VPN 構築 API である VpnService を用いることにより，root 権限不要な実装手法を提案してきた．しかし，VpnService を利用したモデルは，IPv4 ネットワークにおいて IPv4 アプリケーションの利用を想定した実装に留まっており，今後のモバイルインターネットを考慮すると IPv6 対応は必須であった．本稿では，VpnService を用いた NTMobile を IPv6 対応へと拡張を行うことで，非 root 端末において，IPv4/IPv6 各ネットワークで IPv4/IPv6 両アプリケーションのシームレスな接続性を確保する．また，提案方式の実装を行い評価した結果，実用上問題ないスループット特性を得られることを確認した．

1. はじめに

現在，IPv4 グローバルアドレスの枯渇に伴い，次世代規格である IPv6 へ移行が進められている．しかし，IPv4 と IPv6 には互換性がないため，両プロトコル間で相互通信を行うことができない．そのため，既存の IPv4 から次世代の IPv6 のネットワークへの移行を即座に行うことは困難であり，今後当面の間は IPv4 と IPv6 が混在した環境が続くことが想定される．

IPv4 から IPv6 への移行技術として，トランスレータ技術を用いた NAT-PT（Network Address Translation-Protocol Translation）[1] や NAT64 [2]，トンネリング技術を用いた 6to4 [3] や 6rd（IPv6 rapid deployment）[4] などが標準化されている．トランスレータ技術は，IPv4 パケットと IPv6 パケットを相互変換することにより，IPv4 ネットワークと IPv6 ネットワーク間での通信を可能とする．NAT-PT および NAT64 では，IPv4 と IPv6 ネットワークの境目に設置したトランスレータが IPv4 パケットと IPv6 パケットの変換を行うことにより，両ネットワーク間の通信の橋渡しをする．トンネリング技術は，アプリケーションが送信したパケットを IPv4 または IPv6 でカプセル化することにより，IPv4 アプリケーションを IPv6 ネットワーク上で利用することや，IPv6 アプリケーションを IPv4 ネットワーク上で利用することを可能とする．6to4 や 6rd では，IPv4 ネットワークへ接続したルータやエンド端末がデュアルスタックネットワークへ設置したトンネルサーバとの間に IPv4 トンネルを構築することにより，IPv6 ネットワークから IPv4 ネットワークへ接続することができる．

一方，既存の IPv4 ネットワークでは NAT（Network Address Translation）を導入してプライベートネットワークを構築することが一般的であり，CGN（Carrier Grade NAT）[5] のようにキャリアレベルでも NAT が導入され始めている．そのため，多くのユーザは依然として IPv4 プライベートネットワーク上でインターネットを利用しているのが実態である．ところが，NAT が導入された環境においては，グローバルネットワーク側の端末からプライベートネットワーク側の端末に対する接続性を確保できない，NAT 越え問題と呼ぶ課題があり，エンドツーエンドの接続性というインターネット本来の理念を損なう要因となっている．また，既存の IPv6 移行技術は，NAT 配下の端末への通信を考慮していないため，IPv6 ネットワークから IPv4 プライベートネットワークの端末へ通信を行うことができない．今後の IP ネットワークを想定すると，NAT 環境や IPv4 と IPv6 が混在した環境においてもシー

[1] 名城大学大学院理工学研究科
Graduate School of Science and Technology, Meijo University
[2] 愛知工業大学情報科学部
Faculty of Information Science, Aichi Institute of Technology

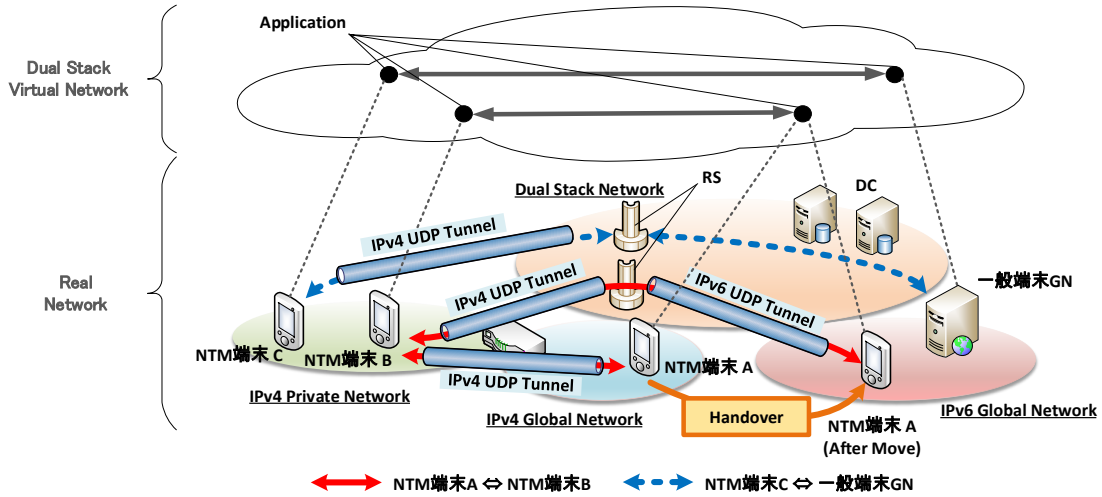

図 1　NTMobile ネットワーク構成

ムレスな接続性を確保する技術が必要である．

筆者らは複雑化したインターネット上に透過的な仮想ネットワークを構築することで，モバイル端末同士が実際のネットワークの違いに影響されることなくシームレスな通信を実現する技術として NTMobile（Network Traversal with Mobility）[6–9] を提案してきた．NTMobile では，NTMobile を導入した端末（以後，NTM 端末）にネットワークの移動によって変化しない仮想 IPv4 アドレスと仮想 IPv6 アドレスを割り当てる．NTM 端末のアプリケーションは，自身の通信プロトコルに対応した仮想 IP アドレスを用いて通信を行うことで，ネットワークの移動による実 IP アドレスの変化を隠蔽し，通信を継続することができる．また，アプリケーションが送信した仮想 IP アドレスに基づくパケットは，NTM 端末間に構築した UDP トンネルを用いて実 IP アドレスによるカプセル化を行う．これにより，実ネットワークに依存しない自由な通信が可能となる．

当初の NTMobile は，OS のカーネルへ機能実装が可能な Linux OS を対象として研究されており，モバイル端末への適用としては，Linux OS がベースである Android OS に実装がされている．しかし，Linux カーネル空間に実装を行っているため，スマートフォンなどのモバイル端末で利用するためには，端末を改造して root 権限を取得する必要があった．root 権限を取得するためには専門的な知識が必要となるほか，root 化した端末はメーカからのアフターケアを受けられないといった弊害が生じる．そのため，一般のモバイルユーザが利用するのは非常に困難であり，NTMobile 普及の大きな壁となっていた．

そこで筆者らは，Android 4.0 以降で利用が可能な，一般のアプリケーションが VPN（Virtual Private Network）を構築する仕組みである VpnService [10] を用いることにより，root 権限なしで NTMobile を実装する手法を提案している [11]．この手法により，一般のモバイルユーザでも手軽に NTMobile を利用することが可能となった．しかし，これまでの実装は IPv4 環境下において IPv4 アプリケーションの利用を想定した実装に留まっており，IPv6 環境および IPv6 アプリケーションへの対応は未実装であった．

本稿では，VpnService を用いた NTMobile の実装を IPv6 対応に拡張することにより，IPv4/IPv6 の各ネットワーク上で IPv4/IPv6 両アプリケーションが接続性を確保するための実装を行った．また，提案方式におけるスループットを測定し，通常の場合の IPv4/IPv6 通信のスループットとの比較を行った結果，提案方式によるスループットの低下は，実用上問題ないことを確認した．

以降，2 章で NTMobile の概要と従来までの実装モデルを紹介し，3 章では提案方式の概要と動作を述べる．4 章で提案方式の実装と評価を行い，5 章でまとめる．

2. NTMobile

2.1 概要

図 1 に NTMobile のネットワーク構成を示す．NTMobile は，NTMobile を実装した NTM 端末と NTM 端末のアドレス情報の管理やトンネル構築の指示を行う DC（Direcition Coordinator），特定の状況下において通信の中継を行う RS（Relay Server）から構成される．DC および RS は IPv4 ネットワークと IPv6 ネットワークのどちらからでもアクセスできるよう，デュアルスタックネットワークに設置し，ネットワークの規模に応じて分散配置することが可能である．

NTM 端末は，端末起動時に自身の FQDN と接続してい

るネットワークから割り当てられた実 IP アドレスを DC に登録する．この時 DC から仮想 IP アドレスを割り当てられ，NTM 端末は仮想 IP アドレスと実 IP アドレスの 2 種類のアドレスを保持する．NTM 端末のアプリケーションは常に割り当てられた仮想 IP アドレスを用いて通信を行うため，実 IP アドレスの変化や通信系路上の NAT の有無，IPv4/IPv6 ネットワークの違いなどによる影響をうけない．また，NTM 端末は定期的に DC と keep alive のメッセージを交換しており，NTM 端末がプライベートネットワークに移動した際にも DC は常に NTM 端末に制御メッセージを送信することができる．

DC は，それぞれ重複のない仮想 IP アドレスプールを保有しており，登録処理の際に仮想 IPv4/IPv6 アドレスを NTM 端末に割り当てる．また，NTM 端末から受け取った FQDN と NTM 端末の実 IP アドレス情報，割り当てた仮想 IP アドレスを対応付けて自身のデータベースに登録する．NTM 端末が通信を開始する際には，データベースに登録してある情報を基に，NTM 端末が接続しているネットワークに応じて最適な経路でトンネル構築指示を行う．

RS は，IPv4/IPv6 ネットワーク間の通信を行う場合や通信相手端末が NTMobile を実装していない一般端末 GN（General Node）である場合など，直接通信ができない状況において通信の中継を行う．

2.2 従来型 NTMobile の課題

当初の NTMobile は，カーネルへ機能実装が可能な Linux OS をインストールした PC を対象に研究されていたが，近年のスマートフォンをはじめとする高性能なモバイル端末の普及に伴い，NTMobile をモバイル端末に適用させたいという要求があった．そのため，NTMobile のモバイル端末への適用として，Linux OS がベースである Android OS に実装を行い，動作検証を行ってきた [6]．しかし，Linux カーネルに NTMobile の機能を加える必要があるため，実装するためには root 権限の取得が必須である．そのため，root 化を行っていない一般のモバイル端末では実装することができず，一般のモバイルユーザへの NTMobile 普及が課題であった．

2.2.1 VpnService 利用型実装モデル

Android OS のバージョン 4.0 以降では，root 権限無しで VPN を構築するための API である VpnService が利用可能となった．VpnService を利用したアプリケーションは，VPN で使用する仮想インタフェースの作成および IP アドレスの設定，ルーティングテーブルの設定と仮想インタフェースに届いた IP パケットのフックすることができる．この API を用いることにより，一般のモバイル端末にも NTMobile の適用を可能とした実装モデルが図 2 に示す VpnService 利用型実装モデル [11] である．

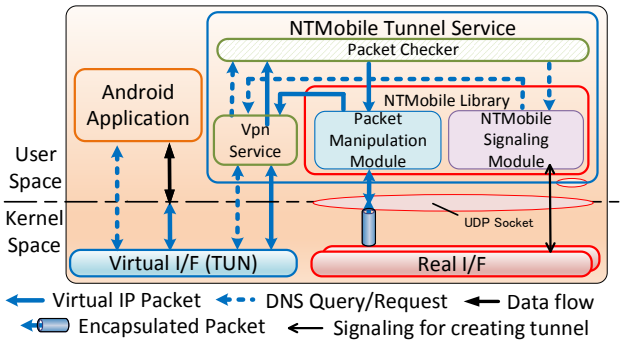

図 2　VpnService 利用型の実装モデルとパケットフロー

VpnService 利用型は，従来 Linux カーネル特有の機能である Netfilter [12] を用いて行っていたパケットのフック処理およびカプセル/デカプセル化処理を VpnService で代用する．また，NTMobile の機能をライブラリとして VpnService を利用した NTMobile のアプリケーション（以後，NTMobile トンネルサービス）に追加する．NTMobile ライブラリは，トンネル構築などの NTMobile のシグナリング処理や，パケットの暗号化/復号処理や送受信などのフックしたパケットのルーティングを行う．

NTMobile トンネルサービスは，Android のサービスとしてバックグラウンドで動作し，一般のアプリケーションが送信した DNS 問合せと仮想 IP アドレス宛のパケットをフックする．フックしたパケットはパケットチェッカーで判別され，DNS 問合せのパケットであると判断すると，NTMobile のシグナリングモジュールにてトンネル構築処理を開始する．トンネル構築が完了すると，自身のトンネルテーブルに経路情報を追加し，取得した CN の仮想 IP アドレスを記載した DNS 応答パケットを生成する．生成した DNS 応答パケットは，DNS 問合せの応答として仮想インタフェースを経由して一般のアプリケーションに渡す．これにより一般のアプリケーションは通信相手を仮想 IP アドレスで認識し，以後はその仮想 IP アドレス宛にパケットを送信する．フックしたパケットが仮想 IP アドレス宛のパケットであれば，単にデータとして扱い，暗号化を行った後に CN の実 IP アドレス宛に UDP ソケットを用いて実インタフェースから送信することで，カプセル化を実現する．以上のようにして，従来の NTMobile がカーネル空間で行っていたパケットのフックおよびカプセル化/デカプセル化処理をユーザ空間のみで実現している．

しかし，これまで VpnService 利用型は，IPv4 環境下において IPv4 通信を行うアプリケーションの利用を想定した実装に留まっており，IPv6 環境および IPv6 対応のアプリケーションにおいて利用することができない．今後のモバイルネットワークを考慮すると IPv6 への対応は必須である．

Vol.2015-DPS-163 No.3
Vol.2015-MBL-75 No.3
2015/5/28

3. 提案方式

3.1 概要

図 3 に提案方式の概要を示す．提案方式では，VpnService 利用型の IPv6 対応として，NTMobile トンネルサービスの機能を拡張し，IPv6 アプリケーションと IPv6 環境へ対応させる．IPv6 アプリケーションに対し NTMobile の通信を実現するためには，アプリケーションが仮想 IPv6 アドレスに基づいたコネクションを確立する必要がある．そのため，NTMobile トンネルサービスに IPv6 アプリケーションが送信した IPv6 パケットのフックおよび解析処理を追加し，IPv6 アプリケーションの DNS 問い合わせに対する AAAA レコードの応答処理を実装する．また，IPv6 環境においてシームレスな通信を実現するためには，NTMobile による IPv6 トンネル通信を行う必要がある．そのため，従来 NTMobile トンネルサービスが使用していたトンネルテーブルを IPv6 対応に拡張し，NTMobile の IPv6 のカプセル化処理を追加する．

この手法を用いることにより，IPv4/IPv6 の各ネットワーク上において IPv4/IPv6 の両アプリケーションは実ネットワークを意識することなく自由な通信を実現できる．すなわち，IPv6 ネットワーク上で IPv4 通信を行ったり，IPv4 ネットワーク上で IPv6 通信を行うことが可能となる．

3.2 動作手順

IPv6 環境下において，NTM 端末である MN（Mobile Node）の IPv6 アプリケーションが通信相手の NTM 端末 CN（Correspondent Node）と通信を開始するまでの動作を図 4 に示す．

3.2.1 登録処理

NTMobile トンネルサービスは，起動時に NTMobile の登録処理を行い，DC より仮想 IPv4 と仮想 IPv6 アドレスを取得する．その後，VpnService を起動し，取得した 2 つの仮想 IP アドレスを割り当てた仮想インタフェースを作成する．また，仮想 IPv4/仮想 IPv6 アドレス宛と DNS 問い合わせのパケットをフックするために，仮想 IPv4/仮想 IPv6 アドレス宛のパケットと DNS サーバ宛のパケットを仮想インタフェースから送信するようにルーティングテーブルを設定する．

3.2.2 パケット解析処理

アプリケーションが送信したパケットをフックすると，IP ヘッダのバージョン，プロトコル番号およびポート番号を解析し，DNS 問い合わせのパケットであるかを判別する．

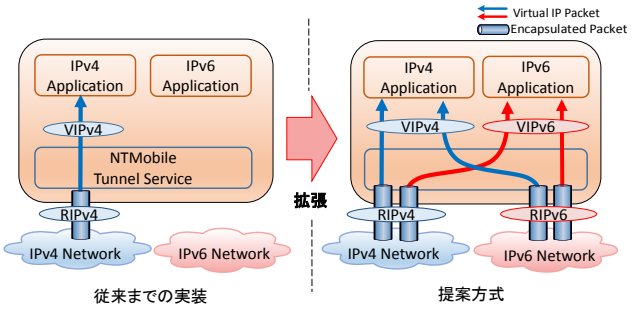

図 3 提案方式の概要

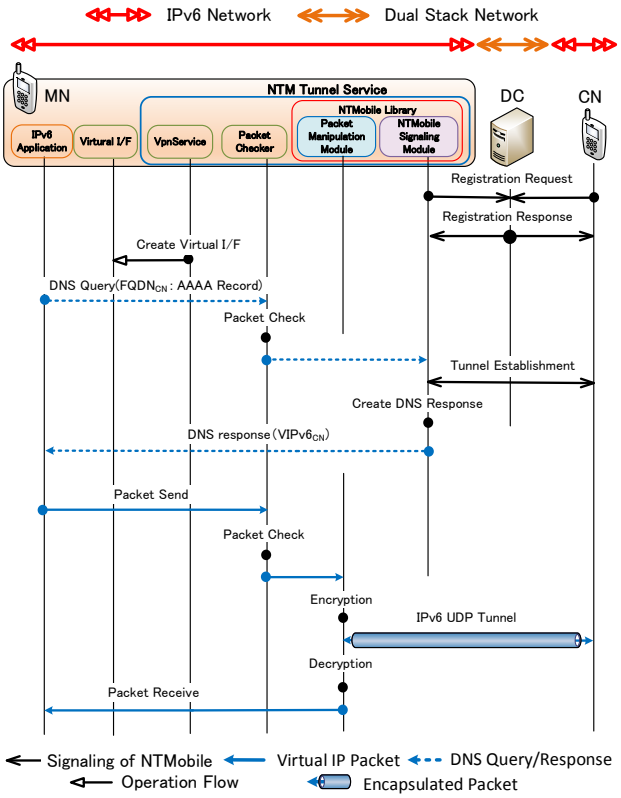

図 4 提案方式の動作シーケンス

3.2.3 名前解決処理

DNS 問い合わせのパケットであると判断すると，$FQDN_{CN}$ と問い合わせレコードを確認した後，NTMobile シグナリングモジュールにてトンネル構築を開始

する．トンネル構築が完了すると，トンネルテーブルと DNS 応答パケットを生成し，DNS 応答パケットを仮想インタフェース経由でアプリケーションに返す．DNS 応答パケットには問い合わせレコードに対応した CN の仮想 IP アドレスが記載されており，IPv4/IPv6 の両アプリケーションは自身の通信プロトコルに対応した仮想 IP アドレスに基づくコネクションを確立する．

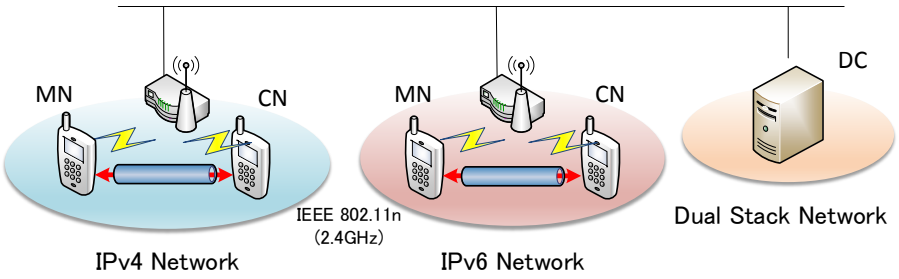

図 5　測定環境

表 1　各装置の仕様

	DC	MN,CN
Hardware	Virtual Machine	Galaxy Nexus
OS	Ubuntu 10.04	Android 4.2.2
Kernel	Linux 2.6.32	Linux 3.0.31
CPU	Intel Core i5-4258U 2.4GHz	Texas Instruments OMAP 4460 1.2GHz
Memory	1GB	1GB

3.2.4　送信処理

フックしたパケットが DNS 問い合わせではないと判断した場合，フックしたパケットを NTMobile ライブラリに渡し，送信処理を開始する．NTMobile ライブラリは宛先の仮想 IPv4 または仮想 IPv6 アドレスを確認し，それを検索キーとしトンネルテーブルを検索する．トンネルテーブルよりトンネル構築相手の実 IP アドレスを取得すると，

暗号化を行った後 UDP ソケットを用いて実インタフェースからカプセル化したパケットを送信する．

3.2.5　受信処理

カプセル化パケットを受信した場合は，NTMobile トンネルサービスがパケットを受信し，デカプセル化および復号を行った後に VpnService を用いて仮想インタフェースに書き込みを行うことで一般のアプリケーションへ渡す．

4.　実装と評価

提案方式を 2 台の Galaxy Nexus（Android 4.2.2，ビルド番号 JDQ39）に実装し，IPv4/IPv6 の各ネットワーク上において IPv4/IPv6 の両アプリケーションの接続性の確認およびスループット評価を行った．

4.1　測定環境

図 5 と**表 1** にネットワーク構成と各装置の仕様を示す．IPv4，IPv6 のネットワーク上にアクセスポイントを設置し，MN と CN は同一のアクセスポイントに接続させた状態で測定を行った．DC は仮想マシン（Ubuntu 10.04）上に構築し，デュアルスタックネットワークに接続させた．

なお，無線環境には IEEE 802.11n（2.4GHz）を用いた．

4.2　測定方法

両端末を IPv4 ネットワークに接続させた場合と IPv6 ネットワークに接続させた場合において，Iperf を用いた 30 秒間の非暗号化 TCP 通信を 10 回行い，そのスループットを測定した．提案方式では，IPv4/IPv6 の各ネットワーク上において IPv4/IPv6 の両アプリケーションが接続性を確保できることを確認するために，各環境下で Iperf による IPv4 と IPv6 の通信を行い，全 4 パターンのスループットを測定した．また，通常の IPv4/IPv6 通信においてもスループットの測定を行い，提案方式との比較を行った．

4.3　測定結果

図 6，**図 7** は IPv4 環境におけるスループット測定の 10 回の平均値とその測定値を示しており，**図 8**，**図 9** は IPv6 環境下におけるスループット測定の 10 回の平均値とその測定値である．また，図中において，General は NTMobile を利用しない通常の通信のスループットであり，NTMobile Tunnel Service は提案方式のスループットを示しており，提案方式の仮想 IPvX を実 IPvY でカプセル化した通信を VIPvX over RIPvY と表す．

測定の結果，IPv4 環境下で最大 8.64%，IPv6 環境下で最大 5.76%程度に収まっていることを確認した．

通常の通信と比較して提案方式のスループットが低下する原因として，パケットフックによるカーネル空間とユーザ空間のパケットの受け渡しの際のメモリコピーや，カプセル化によるヘッダオーバーヘッドが考えられる．

しかし，状態の変化が激しい無線環境での利用を想定した場合，無線状態がスループットに与える影響も大きく，提案方式程度のスループットの低下であれば実用上問題ないと考えられる．

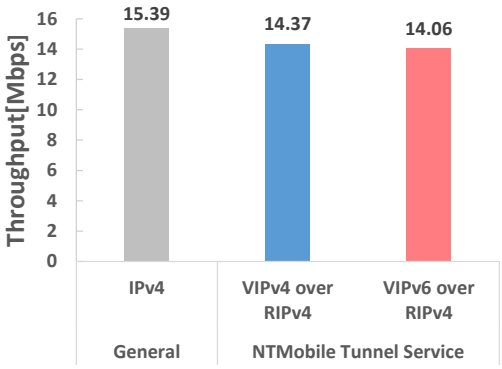

図 6　IPv4 環境下におけるスループット平均値

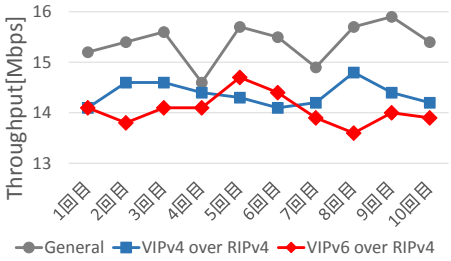

図 7　IPv4 環境下におけるスループット測定値

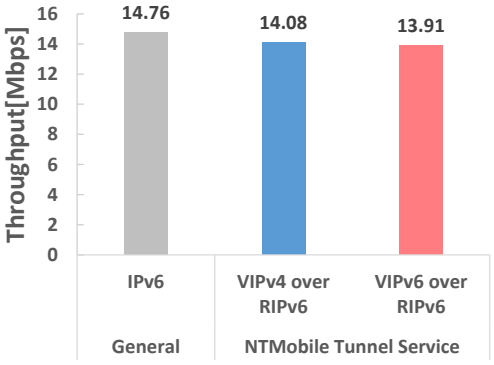

図 8　IPv6 環境下におけるスループット平均値

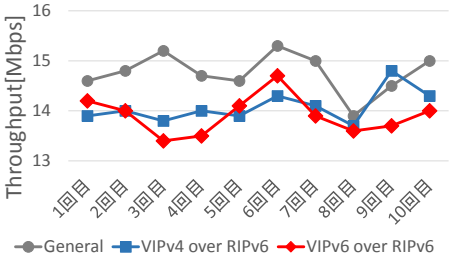

図 9　IPv6 環境下におけるスループット測定値

5. まとめ

本稿では，NTMobile における VpnService 利用型の実装を拡張し，IPv6 に対応させることにより，IPv4/IPv6 の各ネットワーク上で IPv4/IPv6 両アプリケーションに対してシームレスな通信を実現する手法を提案した．提案方式の有効性を評価するため，通常の IPv4/IPv6 通信と提案方式のスループットを測定した．その結果，通常の通信に比べ，提案方式では IPv4 環境下において最大 8.64%，IPv6 環境下において最大 5.76%スループットが低下することを確認した．提案方式により実用上問題ない程度のスループットの低下は発生するものの，Android スマートフォンを root 化することなく，IPv4/IPv6 混在環境で接続性を確保できるという大きなメリットを享受することができる．

参考文献

[1] Tsirtsis, G. and Srisuresh, P.: Network Address Translation - Protocol Translation (NAT-PT), *RFC 2766,IETF* (2000).
[2] Bagnulo, M., Matthews, P. and van Beijnum, I.: Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers, *RFC 6146,IETF* (2011).
[3] Carpenter, B. and Moore, K.: Connection of IPv6 Domains via IPv4 Clouds, *RFC 3056,IETF* (2001).
[4] Townsley, W. and Troan, O.: IPv6 Rapid Deployment on IPv4 Infrastructures (6rd) Protocol Specification, *RFC 5969,IETF* (2010).
[5] Jiang, S., Guo, D. and Carpenter, B.: An Incremental Carrier-Grade NAT (CGN) for IPv6 Transition, RFC 6264, IETF (2011).
[6] 上醉尾一真，鈴木秀和，内藤克浩，渡邊　晃：NTMobile の Android 端末への実装と評価，モバイルコンピューティングとユビキタス通信研究報告， Vol. 62, No. 19, pp. 1–8 (2012).
[7] 上醉尾一真，鈴木秀和，内藤克浩，渡邊　晃：IPv4/IPv6 混在環境で移動透過性を実現する NTMobile の実装と評価，報処理学会論文誌，Vol. 54, No. 10, pp. 2288 – 2299 (2013).
[8] 内藤克浩，鈴木秀和，渡邊　昇，森香津夫，小林英雄：スマートフォンアプリケーションにおいてエンド間通信を実現可能なプラットフォーム開発，マルチメディア，分散，協調とモバイル（DICOMO2014）シンポジウム論文集 (2014).
[9] 鈴木秀和，内藤克浩，渡邊　晃：ユーザ空間における移動透過通信技術の設計と実装，マルチメディア，分散，協調とモバイル（DICOMO2014）シンポジウム論文集，Vol. 2014, pp. 1319–1325 (2014).
[10] VpnService — Android Developers. `http://developer.android.com/reference/android/net/VpnService.html`.
[11] 水野貴文，上醉尾一真，鈴木秀和，内藤克浩，渡邊　晃：スマートフォン向け移動透過通信技術の実装手法に関する提案，情報処理学会全国大会講演論文集，Vol. 76, No. 3 (2014).
[12] Netfilter. `http://www.netfilter.org/`.