

拡張NTMobileを用いたアプリケーションレベルで 実現するシームレス IP Flow Mobility

松岡 穂^{1,a)} 鈴木 秀和^{1,b)} 内藤 克浩²

受付日 2021年4月19日, 採録日 2021年10月8日

概要: 複数の異種通信インタフェースを備えたモバイル端末の普及により, ユーザがモバイルネットワークと Wi-Fi ネットワークを使い分ける機会が増加している. ユーザの移動や通信インタフェースの切替えによりハンドオーバーが発生すると, 全トラフィックが移動先ネットワークへオフロードされる. そのため, 移動先ネットワークの通信状況や品質によってはユーザの体感品質 QoE (Quality of Experience) の低下が懸念される. この課題の解決策として, セッション単位で特定の通信フローをオフロードする IP Flow Mobility 技術がある. 本論文では, 端末主導で移動透過性を実現する技術である NTMobile (Network Traversal with Mobility) を拡張することにより, カーネルの改造が不要でアプリケーションレベルでシームレス IP Flow Mobility を実現する手法を提案する. 提案手法を実装した市販の Android スマートフォンを用いて実環境下で動作検証を行った結果, パケットロスが発生しないシームレスな IP Flow Mobility を実現できることを実証した.

キーワード: IP Flow Mobility, シームレスハンドオーバー, NTMobile, Android

Application-level Seamless IP Flow Mobility with Extended NTMobile

MINORU MATSUOKA^{1,a)} HIDEKAZU SUZUKI^{1,b)} KATSUHIRO NAITO²

Received: April 19, 2021, Accepted: October 8, 2021

Abstract: With the widespread use of mobile devices equipped with multiple heterogeneous communication interfaces, there are increasing opportunities for users to use both mobile networks and Wi-Fi networks. When a handover occurs due to user movement or communication interface switching, all traffic is offloaded to the destination network. Therefore, there is concern that the user's quality of experience (QoE) may decrease depending on the communication status and quality of the destination network. As a solution to this problem, IP Flow Mobility technology, which offloads specific communication flows on a per-session basis, has been proposed. In this paper, we propose a method to achieve seamless IP flow mobility at the application level without kernel modification by extending NTMobile which is a technology to achieve IP mobility by mobile device initiative. We demonstrate that the proposed method can achieve seamless IP flow mobility without packet loss by using a commercial Android smartphone.

Keywords: IP flow mobility, seamless handover, NTMobile, Android

1. はじめに

スマートフォンなどの複数異種通信インタフェースを備えたモバイル端末の普及や, 音楽, ビデオ, ゲームなどの大量データを消費するサービス, SMS (Social Media Service) などの多岐にわたるサービスの浸透により, モバイルデータトラフィックが継続的に増加しており, 2022年には毎月77エクサバイトの通信量が発生することが予測

¹ 名城大学大学院理工学研究科
Graduate School of Science and Technology, Meijo University, Nagoya, Aichi 468-8502, Japan

² 愛知工業大学情報科学部
Faculty of Information Science, Aichi Institute of Technology, Toyota, Aichi 470-0392, Japan

a) minoru.matsuoka@ucl.meijo-u.ac.jp

b) hsuzuki@meijo-u.ac.jp

されている [1]. この対応策の1つとして, Wi-Fi オフロードイングがあげられ, モバイルと Wi-Fi を適宜切り替えながら通信を行う機会が増加している.

異種ネットワーク間の切替えにおいて, モバイル端末の IP アドレスが変化してしまうため, セッションを維持できず通信断絶が発生するという課題が存在する. この課題を解決するために移動透過性技術があり, これまでに Mobile IPv4 [2], Mobile IPv6 [3], DSMIPv6 (Dual Stack Mobile IPv6) [4], PMIPv6 (Proxy Mobile IPv6) [5], NTMobile (Network Traversal with Mobility) [6], [7], [8] などが提案されてきた. また, IEEE 802.21 [9] と移動透過性技術を連携させることにより, 通信断絶前に移動先ネットワークへの接続処理を実行することにより, パケットロスのないシームレスハンドオーバーを実現することができる [10].

しかし, モバイル端末の移動や接続先ネットワークの切替えにより, 全トラフィックが移動先ネットワーク側にオフロードされる. そのため, 移動先ネットワークの通信状況や通信品質によっては, VoIP (Voice over IP) やビデオ会議, VR (Virtual Reality) などのリアルタイム性を要求する通信フローなどの体感品質 QoE (Quality of Experience) の低下が生じる可能性が考えられる. また, 実運用が始まった第5世代移動通信システム (5G) を介して超低遅延通信を行っている最中に, 街中の Wi-Fi ネットワークにハンドオーバーしてしまうと, 通信品質の低下が予想される. IoT (Internet of Things), AI (Artificial Intelligence), ロボットなどの先端技術をあらゆる産業や社会生活に取り入れ多様な需要にきめ細かく対応したモノやサービスを提供する社会 Society 5.0 [11] を実現するためには, 各通信ネットワークの特性と今後多岐にわたる通信性能要求を最適化することは肝要であるといえる.

前記の課題を解決する技術の1つとして, 3GPP (3rd Generation Partnership Project) で議論されている IP Flow Mobility があげられる [12]. IP Flow Mobility はモバイルネットワークと Wi-Fi ネットワークを同時に利用することにより, セッション単位で特定の通信フローをオフロードする技術であり, 大きく分けてネットワーク主導型と端末主導型のアプローチが存在する [13]. ネットワーク主導型 IP Flow Mobility には PMIPv6 をベースとした手法 [14], [15], [16] などが, 端末主導型には MIPv6 をベースとした手法 [17] が提案されているが, ネットワーク装置がモバイル端末の接続切れを検知してから通信フローの再割当てを行うため, 通信断絶が発生する. また, モバイル端末のカーネルを改造しなければならず, 一般ユーザの利用を考えると大きな障壁であるといえる.

本論文では, IPv4/IPv6 混在環境など様々なネットワークに対してハンドオーバー可能な NTMobile をユーザ空間で拡張することにより, カーネルの改造を不要とした通信断絶の発生しない端末主導型 IP Flow Mobility を実現する手

法を提案する. 従来の NTMobile に拡張メッセージを定義し, 複数の通信インターフェースを用いて通信相手端末との間に複数のトンネル構築が可能なシグナリングを行うように拡張する. 拡張 NTMobile 実装端末は通信断絶前にハンドオーバーを検知し, 追加トンネル構築/該当トンネル利用停止処理を実行し, 完了後, ルーティングポリシーに基づいて通信フローを各トンネルに対して割り当てることにより, IP Flow Mobility を実現する. AWS EC2 (Amazon Web Services Elastic Computer Cloud) 上に各 NTMobile サーバを構築し, 拡張 NTMobile を実装した Android スマートフォンを用いて LTE と Wi-Fi の両ネットワークに接続してハンドオーバーをした結果, パケットロスが発生しない IP Flow Mobility が実現できることを確認した.

以下, 2章で IP Flow Mobility の関連研究とその課題を整理し, 3章で提案手法について述べる. 4章で実装アプローチおよびモジュール構成を示し, 5章で性能評価について述べた後, 6章でまとめる.

2. 関連研究

2.1 ネットワーク主導型 IP Flow Mobility

Purohithらは PMIPv6 と SDN (Software Defined Networking) を組み合わせた SIFM (Seamless Internetnetwork Flow Mobility) アーキテクチャにより IP Flow Mobility を実現している [16]. SIFM アーキテクチャは SDN における FC (Flow Controller), LTE システムの EPC (Evolved Packet Core) 上にある PGW (Packet Gateway), Wi-Fi アクセスポイントを束ねる WAG (Wireless Access Gateway) で構成され, SDN における制御プレーンとデータプレーンに分離する側面を利用し, 従来のモビリティ機能およびフローモビリティ決定機能を FC に移すことにより, PGW の複雑さを軽減するアーキテクチャである (図 1).

PGW と WAG は PMIPv6 における MAG の機能と SDN におけるフロースイッチの機能を持ち, PGW と WAG の間には IP-in-IP トンネルが事前に構築される. FC は両ネットワークの負荷バランスをとることを目的としてリンク

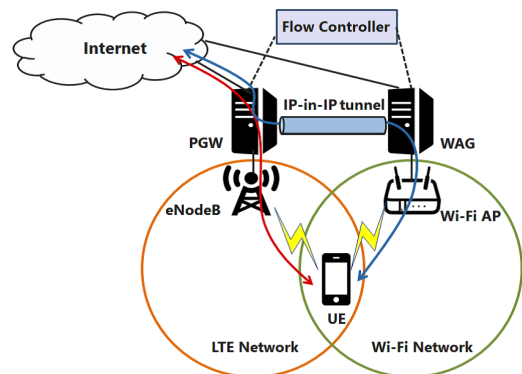


図 1 PMIPv6 と SDN を組み合わせた SIFM アーキテクチャ
Fig. 1 SIFM architecture combining PMIPv6 and SDN.

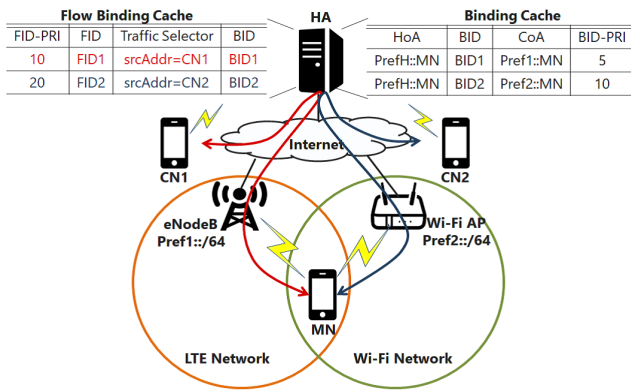


図 2 Mobile IPv6 を拡張した手法の概要

Fig. 2 Overview of methods for extending Mobile IPv6.

の帯域情報、パケットやポートの統計情報などに基づいてフロー割当てを決定する。PGW および WAG は FC によるフロー割当て指示に基づいて動作し、フローは PGW と WAG 間のトンネルを経由して Wi-Fi ネットワークに送信されるか、トンネルを経由することなく LTE ネットワークに送信されるか制御される。

PMIPv6 をベースとしているため、UE は市販のスマートフォンがそのまま利用できるが、携帯電話事業者が上記専用装置を導入および運用するネットワーク内での移動しかサポートされない。そのため、接続先ネットワークによってはフローの割当て以前に通信の継続自体ができない。また、UE の移動により接続が失われたことを PGW もしくは WAG が検知した後に FC によるフロー割当て変更が行われるため、パケットロスが発生し、QoE の低下を招くことが課題となる。

2.2 端末主導型 IP Flow Mobility

Verga らはホームアドレス HoA (Home of Address) に対して複数の気付アドレス CoA (Care of Address) 登録をサポートする RFC5648 [18] と、通信フローと HoA および CoA の組合せを対応づけることでフロー振り分けを可能にする RFC6088 [19] に基づいて拡張した Mobile IPv6 により IP Flow Mobility を実現している [17]。図 2 にこの手法の概要を示す。複数の CoA 登録のサポートでは、従来の Mobile IPv6 における Binding Update メッセージとともに、新たに Binding Identifier Mobility オプションとして CoA に関連付けてバインディング識別子である BID (Binding ID) をホームエージェント HA (Home Agent) に通知し、各バインディングを一意に識別できるよう拡張される。また、各バインディング重複において最も優先度の高いものを選択するためにバインディング優先度である BID-PRI も通知される。このオプションにより、Binding Update を受け取る HA は 1 つの CoA に対して複数の CoA を登録することが可能となる。

Flow Binding のサポートでは Binding Update メッセージ

とともに、新たに Flow Identification Mobility オプションとして端末が HA に対してインバウンドパケットをどのネットワークに対してルーティングするか指定する情報が含まれる。Flow Identification Mobility オプションには BID に対するフロー識別子である FID、FID に関するフロー情報を示すトラフィックセクタ [20] および一意に識別するために利用されるフロー優先度 FID-PRI が含まれる。モバイル端末 MN (Mobile Node) から指示を受けた HA もしくは通信相手端末 CN (Correspondent Node) は新たに FBC (Flow Binding Cache) と呼ばれるデータ構造にフロー振り分け方法を保存する。これにより、フロー振り分けが可能となる。

端末主導型は MN に移動透過性技術を導入し、通信事業者やサービスプロバイダ、個人ユーザなどが HA を運用する必要はあるが、接続先ネットワークによらず通信を継続できるという利点がある。しかし、この手法では MN の Linux カーネルへの機能実装が必須であるため、カーネルに実装する独自モジュールにバグや不具合が混入していた場合、カーネルがハングアップしてスマートフォンが利用できなくなる危険性や、カーネルのバージョンアップにともなう独自モジュールの動作検証や改修をしなければならず、継続的なモジュール提供が困難である。また、一般ユーザが所有する市販の Android スマートフォンに適用することは困難という課題が存在する。また、この手法はハンドオーバー時におけるパケットロスについて言及されていないが、Mobile IPv6 をベースとしているため、RFC5568 による高速ハンドオーバー (FMIPv6) [21] を適用することによりシームレスハンドオーバーを実現できる可能性がある。ただし、MN の移動前および移動先ネットワークの両方に設置されている AR (Access Router) が FMIPv6 対応でなければならず、シームレスハンドオーバーの可否はネットワーク環境に依存してしまう。

2.3 NTMobile

筆者らは端末主導型 IP Mobility 技術として NTMobile を提案している [6], [7], [8]。NTMobile は、NTMobile を実装した端末 (NTM 端末)、NTM 端末のアドレス情報の管理やトンネル構築指示を行う DC (Direction Coordinator)、IPv4-IPv6 間のように直接通信ができない場合に NTM 端末間の通信を中継する RS (Relay Server) により構成される (図 3)。DC および RS は IPv4/IPv6 どちらのネットワークからでもアクセスできるデュアルスタックネットワークに配置される。

NTM 端末は起動時に自身の FQDN と実 IP アドレスを DC に登録する。登録完了後、DC は NTM 端末に対してネットワークの移動によって変化しない仮想 IP アドレスを割り当てる。NTM 端末は DC と定期的に Keep Alive 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

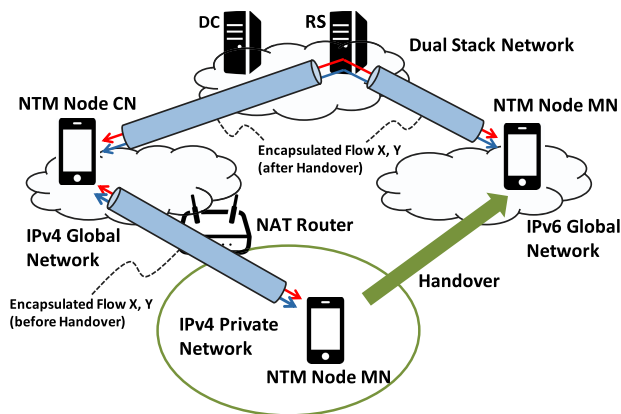


図 3 NTM Mobile の概要
Fig. 3 Overview of NTM Mobile.

ワークに移動した場合にもつねに DC から NTM 端末に制御メッセージを受信できる状態を維持する。

NTM 端末間の通信は仮想 IP アドレスに基づいてエンドツーエンドで暗号化される。仮想 IP アドレスで示されたパケットを実環境でルーティングするために、実 IP アドレスを指定した IP/UDP ヘッダによりカプセル化し、通信相手へ送られる。NTM 端末がハンドオーバーにより接続先ネットワークを切り替えた場合、実 IP アドレスが変化するため、DC に対して新しい実 IP アドレスを通知し、仮想 IP アドレスとの対応関係を更新する。これにより、NTM 端末の FQDN から現在割り当てられている実 IP アドレスおよび仮想 IP アドレスを検索することが可能となる。また、ハンドオーバーした NTM 端末は通信相手 NTM 端末との間でトンネルの再構築処理を行うことにより、再構築した暗号化 UDP トンネル内に移動前に実施していた仮想 IP アドレスに基づく IP パケットを流し込む。これにより、NTM 端末間の通信は維持される。

NTM Mobile は IP レベルの移動透過性を実現するため、モバイル端末の Linux カーネルへの機能実装が必要であるが、VPN フレームワークを活用することにより、カーネルの改造を必要とせず Android スマートフォンへ実装することができる [22]。また、IEEE 802.21 を連携させることにより、通信断絶前に移動先ネットワークへの接続処理を実行するなどの対策を行うことにより、シームレスハンドオーバーを実現している [10]。しかし、ハンドオーバー時にはすべてのセッションを新しいトンネル経路に切り替えてしまうため、IP Flow Mobility の実現には至っていない。

3. 提案手法

3.1 IP Flow Mobility 実現のアプローチ

2 章で述べたとおり、市販の Android スマートフォンの Linux カーネルを改造することなく、移動先ネットワークの制約を受けないシームレスな IP Flow Mobility を実現している例はない。そこで、本論文では NTM Mobile を拡張す

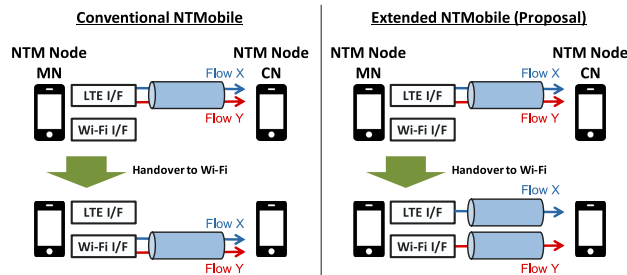


図 4 従来手法と提案手法におけるトンネルとフローの比較
Fig. 4 Comparison of tunneling and flow in conventional and proposed methods.

ることにより IP Flow Mobility を実現するアプローチを採用する。IP Flow Mobility を実現するためには、フローの識別機能およびポリシールーティング機能が必要となる。従来の NTM Mobile はこれらの機能を有していないため、下記の拡張を行う。

- NTM 端末が LTE および Wi-Fi 両方のネットワークに接続しているマルチホーム状態の場合、各通信インタフェース経由で複数のトンネル経路を構築できるようにプロトコルを拡張する。
- NTM Mobile アプリケーション内部に新規構成したルーティングポリシーに基づいて、フローを LTE 側、Wi-Fi 側どちらのトンネル経路に振り分けるか決定し、iproute2 により送信元 IP プレフィックスを条件としてルーティングテーブルを参照する機能を NTM 端末に追加する。
- DC は NTM 端末の LTE 側、Wi-Fi 側両方の IP アドレス情報を管理し、NTM 端末に対して複数のトンネル構築を指示できるように拡張する。

3.2 概要

図 4 に従来の NTM Mobile と本論文で拡張する NTM Mobile を適用した NTM 端末間のトンネルとフローの関係を示す。従来の NTM Mobile^{*1}は LTE、Wi-Fi の通信インタフェースを同時に利用できないため、LTE から Wi-Fi へハンドオーバーする場合、Wi-Fi 側でトンネル再構築処理が完了するまでの間、パケットロスが発生し、すべてのフローが再構築後のトンネルで処理される。これに対して、提案手法ではマルチホーム状態をサポートするため、LTE から Wi-Fi へハンドオーバーする際、LTE 側で通信を継続しながら Wi-Fi 側でトンネルを再構築することができ、パケットロスが発生しない。また、LTE 側、Wi-Fi 側のトンネル経路を同時に利用することができ、かつルーティングポリシーに基づいて各フローを適切なトンネルに振り分けることにより、IP Flow Mobility を実現する。

^{*1} IEEE802.21 対応版は NTM 端末はカーネルの改造が必要なたため、以後、この機能を適用しない NTM Mobile を従来手法と見なす。

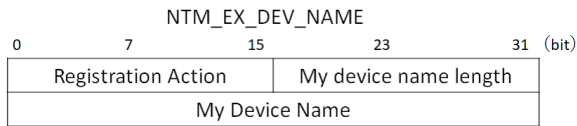


図 5 Registration Request に付与される拡張データ

Fig. 5 Extended data to be added to Registration Request.

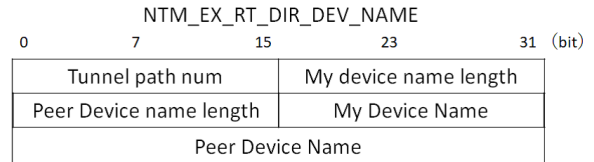


図 6 Route Direction に付与される拡張データ

Fig. 6 Extended data to be added to Route Direction.

3.3 NTMobile の拡張

3.3.1 プロトコルの拡張

NTMobile は NTM 端末における DNS クエリの実行をトリガにトンネル構築シグナリングを開始する。NTMobile 制御パケットのフォーマットは IP ヘッダ、UDP ヘッダ、NTMobile ヘッダ、NTMobile 拡張ヘッダ、シグナリングメッセージ、MAC (Message Authentication Code) の順で構成されている。拡張ヘッダを利用して拡張データを追加することにより、従来の NTMobile と互換性を維持しつつ、新しい機能を追加することが可能である。本論文ではアドレス登録時およびトンネル経路指示時に関する制御メッセージに対して、拡張データを新たに定義する。

アドレス情報登録

従来の NTMobile では、NTM 端末が現在の IP アドレスを DC へ登録するために Registration Request を送信し、端末起動時などの初回登録は新規エントリ作成を、ハンドオーバー時は作成済みエントリに対して上書きを行う。この際、通信インタフェースの違いは考慮しておらず、DC は NTM 端末の IP アドレスとだけ認識している。

これに対して、提案手法では NTM 端末はマルチホーム状態をサポートするため、LTE 側、Wi-Fi 側の双方のネットワークから IP アドレスが割り当てられる。そのため、DC が NTM 端末の複数の IP アドレス情報を管理できるようにするために、NTM 端末は Registration Request に対して通信デバイス名を記載した拡張データ (NTM_EX_DEV_NAME) を付与することにより、DC に LTE と Wi-Fi のアドレス情報を区別させてアドレス登録を行うよう拡張する (図 5)。

Registration Action には定数 ADD, DELETE, HANDOVER が設定され、ADD の場合は新規エントリ作成または上書きを行う。DELETE の場合は NTM 端末が切断予定であるネットワークにおけるアドレス情報の削除を行い、HANDOVER の場合はすでに登録された 1 つのエントリに対して上書きを行う。My Device Name には登録したい IP アドレスが割り当てられている通信インタフェース名が設定される。

トンネル経路指示の拡張

NTMobile は通信開始時にトンネル構築シグナリングを実行するが、どのようなパスで構築するかについては DC が両エンド NTM 端末のアドレス情報に基づいて決定し、NTM 端末および RS を中継する場合は RS にもトンネル経路の指示を行う。従来の NTMobile では、NTM 端末は互いにどの通信インタフェース間でトンネルを構築している

か意識していない。これに対して、提案手法では NTM 端末は自身だけでなく通信相手の通信インタフェースを考慮していくつトンネルを構築すればよいのかを把握する必要がある。そこで、トンネル経路指示に関する制御メッセージ Route Direction に対して、互いの通信インタフェース名や構築すべきトンネル経路の本数などを記載した拡張データ (NTM_EX_RT_DIR_DEV_NAME) を付与して通知できるように拡張する (図 6)。

3.3.2 NTM 端末の拡張

NTM 端末は 3.3.1 項に示した拡張した NTMobile プロトコルに対応するために、制御メッセージに記載された拡張ヘッダの有無を確認し、前述の拡張データの付与、取得および当該処理を実行できるよう拡張される。たとえば、拡張データ NTM_EX_RT_DIR_DEV_NAME が付与された拡張 Route Direction に記載されたトンネル経路数に基づいて、NTM 端末はトンネル構築スレッドを複数立ち上げ、その後に行う Tunnel Request/Response のやりとりを同時に実施できるように拡張する。これにより、トンネル構築に関わるオーバーヘッドの発生を最小化する。

また、NTM 端末は通信インタフェースを指定することにより、使用するトンネルのスイッチングを実現する。MN は CN や RS と複数のトンネルを構築するため、各トンネルに関連付けられたデータグラムソケットを複数管理することになる。これらのソケットを切り替えるパターンとして、トンネル構築を行う Tunnel Request/Response の送信に関する場合と、追加トンネル構築完了後のフロー振り分けに関する場合がある。前者は拡張 Route Direction により使用する通信インタフェース名が通知されることで実現される。後者はフローに対する使用する通信インタフェース名が記載されたルーティングポリシーを参照することで実現される。LTE 用、Wi-Fi 用のソケットは各通信インタフェースに割り当てられている IP アドレスがバインドされ、またルーティングポリシーに従って仮想 IP アドレスが設定されたアプリケーションフローに関する送信パケットをどちらのソケットに対して書き込むかを決定し、選択された通信インタフェースを通じて接続しているネットワークへカプセル化されたアプリケーションパケットが送信される (図 7)。

3.3.3 DC の拡張

DC も NTM 端末と同様に拡張した NTMobile プロトコ

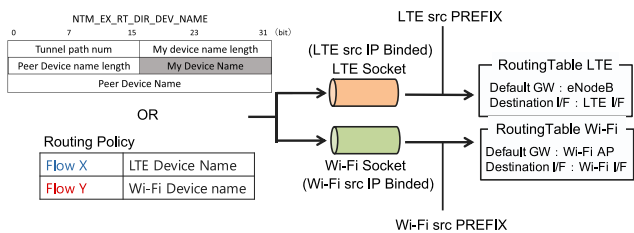


図 7 ソケットスイッチングによるトンネル経路の選択
 Fig. 7 Tunnel route selection by socket switching.

表 1 拡張 Node Address Table
 Table 1 Extended Node Address Table.

id	node_id	dev_name	rip4	rip6	nat_port
1	nodeA	wlan0	rip4_wifi	rip6_wifi	portA
2	nodeA	rmnet_data0	rip4_lte	rip6_lte	portB

ルに対応するように処理の一部と管理するデータを拡張する。DC は Registration Request を受信すると、制御メッセージに記載されている NTM 端末の実 IPv4/IPv6 アドレスに加えて、受信パケットの送信元 IP アドレスを確認することにより、NAT の通過有無を判定し、NAT にマッピングされたグローバル IP アドレスおよびポート番号などの情報を Node Address Table に記録する。提案手法では Node Address Table を拡張し、NTM 端末のどちら側の通信インタフェースで通知されたアドレス情報であることを管理できるように dev_name の項目を追加定義する (表 1^{*2})。これにより、NTM 端末から送信された拡張 Registration Request を受信することにより、NTM 端末のアドレス情報を複数管理できるようにする。

このほか、従来の NTMobile では構築するトンネルの識別子である Path ID を NTM 端末側で指定していたが、提案手法では複数構築されるトンネルを管理しやすくするために DC 側で設定するよう変更し、同一通信経路の場合は同一 Path ID となるように仕様の変更を行う。

3.4 ハンドオーバーシーケンス

提案手法では MN だけでなく、CN もスマートフォンであることを想定している。そのため、両 NTM 端末が LTE と Wi-Fi の通信インタフェースを有しており、それぞれシングルホーム (LTE または Wi-Fi のいずれかのネットワークのみに接続)、マルチホーム (LTE および Wi-Fi の両方のネットワークに接続) の状態がある。このような状況で MN がハンドオーバーするため、提案手法は表 2 に示す 8 通りの通信パターンが想定される。本論文では MN がマルチホーム状態へ遷移する場合とマルチホーム状態からシングルホーム状態へと遷移する場合のパターン 1~4 を中心に取り上げて、ハンドオーバーシーケンスを示す。

表 2 提案手法において想定する通信パターン

Table 2 Communication patterns assumed in the proposed method.

Pattern	MN's connection status		CN's connection status
	Before move	After move	
1	LTE/Wi-Fi	Multi-homed	LTE/Wi-Fi
2	Multi-homed	LTE/Wi-Fi	LTE/Wi-Fi
3	LTE/Wi-Fi	Multi-homed	Multi-homed
4	Multi-homed	LTE/Wi-Fi	Multi-homed
5	LTE	Wi-Fi	LTE/Wi-Fi
6	Wi-Fi	LTE	LTE/Wi-Fi
7	LTE	Wi-Fi	Multi-homed
8	Wi-Fi	LTE	Multi-homed

3.4.1 パターン 1・パターン 3

追加ネットワーク接続および追加アドレス登録

図 8, 図 9 にパターン 1 およびパターン 3 におけるハンドオーバーシーケンスを示す。MN は CN に対して LTE で RS 経由のトンネル構築を構築しており、Flow X, Y の通信を行っているものとする。MN が移動して LTE に加えて Wi-Fi に接続した場合、MN は移動先ネットワーク状況確認として Wi-Fi の RSSI および DC_{MN} との RTT を一定時間測定する。その結果、RSSI が閾値以上、RTT が閾値以下であり続けた場合、MN は移動先ネットワークである Wi-Fi を利用して DC_{MN} に対して拡張 Registration Request を送信し、Wi-Fi アドレス登録処理を行う。

追加トンネル構築

アドレス登録を完了した MN は、LTE 側で Flow X, Y の NTMobile トンネル通信を継続しながら、Wi-Fi 側から DC_{MN} に対して Direction Request を送信し、トンネル構築処理を行う。トンネル経路指示要求を受けた DC_{MN} は DC_{CN} と Node Information Request/Response を行い、CN のアドレス情報を取得してトンネル経路を決定する。その後、DC_{MN} はトンネル経路が RS 経由と決定した場合は RS に対して Relay Direction を送信し、応答として NTM ACK を受信した後、MN および CN (DC_{CN} 経由) へ複数の拡張 Route Direction を送信することにより、トンネル構築経路指示を行う。ここで、DC_{MN} は MN および CN に対して、それぞれが使用している無線インタフェースに対して経路指示を行うことにより、考えられるすべての組合せのトンネルが構築される。経路指示を受けた MN および CN は複数の Tunnel Request/Response をやりとりし、トンネル追加構築処理を完了する。なお、パターン 1 とパターン 3 の差異はトンネル構築本数が 2 本となるか 4 本となるかの違いである。

フロー振り分け

トンネル追加構築が完了した MN は、CN に対する RS 経由の 2 本のトンネルに対して通信フローを振り分ける。MN は通信フローに対してどの通信インタフェースを利用

*2 主要なフィールドのみを示し、フィールド値は簡単な表現にて示されている。

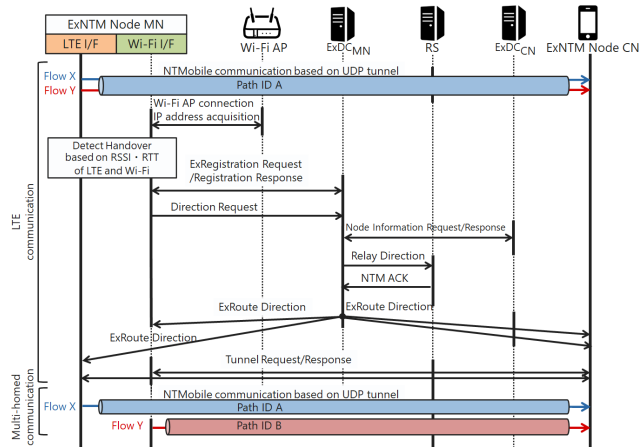


図 8 パターン 1 (MN : LTE からマルチホーム, CN : シングルホーム) のハンドオーバーシーケンス

Fig. 8 Handover sequence in case of pattern 1 (MN: LTE to multi-homed, CN: single-homed).

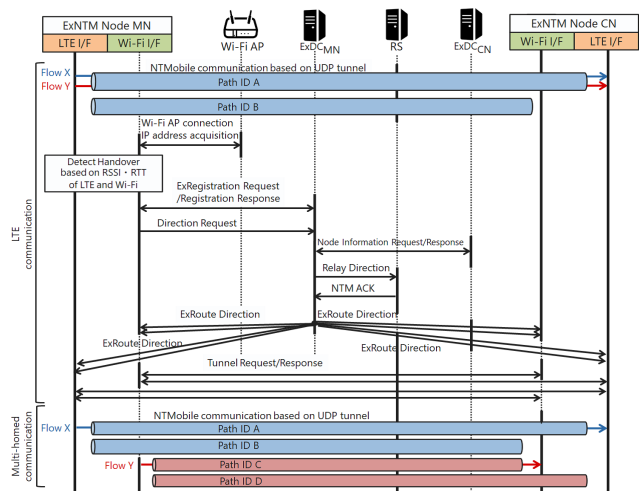


図 9 パターン 3 (MN : LTE からマルチホーム, CN : マルチホーム) のハンドオーバーシーケンス

Fig. 9 Handover sequence in case of pattern 3 (MN: LTE to multi-homed, CN: multi-homed).

するか記載されたルーティングポリシーに従い、通信フローを振り分ける。ここでは、Flow X を MN から見て LTE 側のトンネルに、Flow Y を MN から見て Wi-Fi 側のトンネルに振り分けたものとする。なお、MN と CN がお互いにマルチホーム状態であるパターン 3 の場合は合計 4 本のトンネルが構築されることになるが、LTE どうしまたは Wi-Fi どうしで構築された 2 本のトンネルのどちらかを使用する。なお、残りの 2 本のトンネルは、MN または CN がシングルホーム状態になったときに使用する可能性がある。事前にトンネルを構築しておくことにより、ハンドオーバー時に即座にフローを切り替えることができる。

3.4.2 パターン 2・パターン 4

図 10, 図 11 にパターン 2 およびパターン 4 におけるハンドオーバーシーケンスを示す。ここでは、MN がマルチホームから Wi-Fi にハンドオーバーした場合について述べ

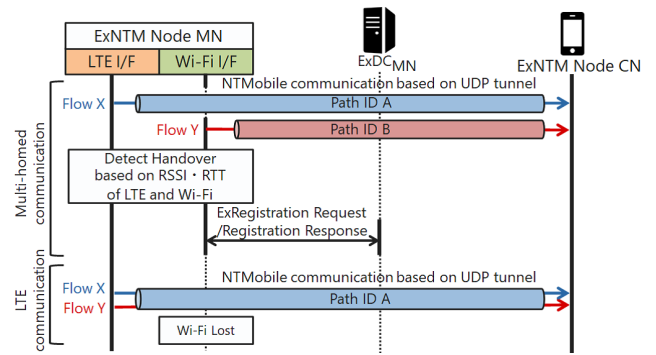


図 10 パターン 2 (MN : マルチホームから LTE, CN : シングルホーム) のハンドオーバーシーケンス

Fig. 10 Handover sequence in case of Pattern2 (MN: Multi-homed to LTE, CN: Single-homed).

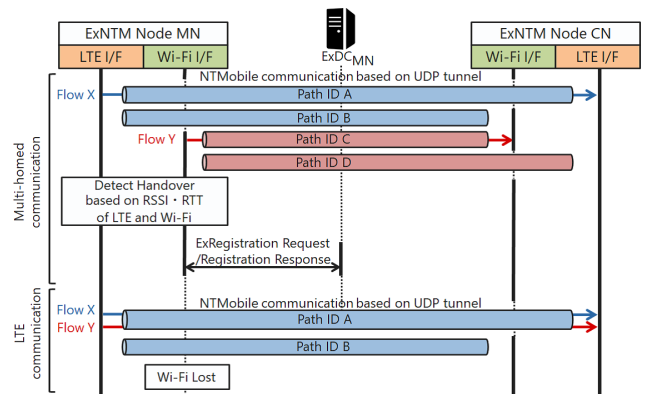


図 11 パターン 4 (MN : マルチホームから LTE, CN : マルチホーム) のハンドオーバーシーケンス

Fig. 11 Handover sequence in case of Pattern4 (MN: Multi-homed to LTE, CN: Multi-homed).

る。一定時間の間、Wi-Fi の RSSI が閾値以下であり続けたこと、または DC_{MN} との RTT が閾値以上であり続けたことを検知した MN は、Wi-Fi 側のトンネルキープアラートを利用して CN に対して MN 自身の Wi-Fi が接続終了することを通知する。また MN は DC_{MN} に対して、Wi-Fi 経由で拡張 Registration Request を送信し、Wi-Fi の接続終了を通知する。 DC_{MN} は接続終了通知を受信すると、拡張 Node Address Table で管理していた MN の Wi-Fi アドレス情報を削除する。MN は通知後、すべての通信フローを LTE トンネルに割り当て、通信断絶前にハンドオーバーを完了する。なお、パターン 2 とパターン 4 の違いはトンネル構築数の違いのみである。

これにより、従来の NTMobile のようなハンドオーバー用のトンネル構築シーケンスにおける Route Direction, Tunnel Request/Response を排した即時切替が実現される。

4. 実装

本論文では Android の VpnService を用いた NTMobile [22] を拡張することにより、提案手法の各機能をアプリケーションレベルで実装する。なお、今回のプロト

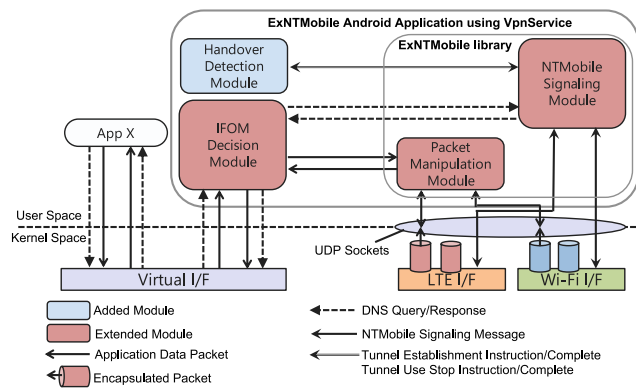


図 12 提案手法を適用した Android 端末のモジュール構成
Fig. 12 Module configuration of an Android device applying the proposed method.

タイプ実装はアプリケーションレベルでシームレスな IP Flow Mobility を実現できることを実証することを目的としており、実運用で必要な機能については省略した。

4.1 拡張 NTM 端末のプロトタイプ実装

図 12 に Android スマートフォンに拡張 NTMobile を実装した際のモジュール構成を示す。拡張した NTMobile アプリケーションは、Handover Detection Module (HDM), NTMobile Signaling Module (NSM), IFOM Decision Module (IDM), Packet Manipulation Module (PMM) の計 4 つのモジュールで構成される。

4.1.1 Handover Detection Module (HDM)

HDM は既存の NTMobile に対して追加されたモジュールであり、Kotlin を用いて実装した。NTM 端末の接続状態把握およびマルチホーム時にハンドオーバー検知を行い、必要に応じてトンネル構築指示/該当トンネル使用停止指示を NSM に対して行うモジュールである。

ハンドオーバー検知タスクでは、Android OS より LTE および Wi-Fi の RSSI を取得し、RTT に関しては自身を管理する DC に対して ping コマンドを実行することで取得する。RSSI と RTT は一定間隔で取得され、3.4 節で述べた条件が満たされると、HDM は NSM に対して追加接続したネットワークを利用してトンネルを追加構築するよう指示を行う。なお、この指示を行う際、Android 端末に標準搭載されている iproute2 による ip rule コマンドを実行して、送信元プレフィックスよりカプセル化パケットが適切なルーティングテーブルを参照するよう設定を行う。

4.1.2 NTMobile Signaling Module (NSM)

NSM は拡張 NTMobile ライブラリに該当し、C 言語で記述されている。NTMobile シグナリングを行うモジュールであり、HDM の指示を受けて、指定通信インタフェースによる DC へのアドレス追加登録や CN へのトンネル追加構築を実行することが可能であるように拡張されている。

4.1.3 IFOM Decision Module (IDM)

IDM は仮想 IP パケットをフックし、パケット解析を行い、DNS クエリである場合は NSM にトンネル構築を指示し、アプリケーションのデータパケットである場合は PMM にパケットの送信指示を行うモジュールである。IDM はルーティングポリシーを管理し、これは Room Persistence Library^{*3} (以後、Room) を用いて、フロー情報^{*4}、使用通信インタフェース、Outgoing フラグ、Incoming フラグフィールドを持つデータベースとして実装される。

端末がマルチホーム状態となったとき、フックした仮想 IP パケットから取得したフロー情報を用いてルーティングポリシーを検索する。ヒットした場合、Incoming フラグが立っていない場合は Outgoing フラグを立て、使用通信インタフェースとパケットを PMM に渡す。通信相手から仮想 IP パケットを受信した場合、解析しフロー情報からルーティングポリシーを検索するが、もし Outgoing フラグが立っていない場合は Incoming フラグを立て、通信相手の通信インタフェースで使用通信インタフェースフィールドを更新する。これら機能を Kotlin で実装することにより、他の IP Flow Mobility 手法においても言及されている対称通信経路を実現する。

今回のプロトタイプ実装では、ルーティングポリシーは動作検証用として事前にいくつかの情報を設定した。

4.1.4 Packet Manipulation Module (PMM)

PMM はトンネルテーブルを参照して、カプセル化パケットの送信および受信処理および暗号化処理などを行うモジュールである。暗号化処理などを高速に実行するために、C 言語で実装された NTMobile ライブラリの一部として実装される。提案手法では適切な通信インタフェースを使ってトンネル通信を行えるようにするために、マルチホーム時に IDM から指定された通信インタフェースで構築されたトンネルを使用してパケットを送信するように拡張した。トンネルテーブルのエントリは DC による経路指示を受けて作成されるが、複数トンネル経路構築のための通信デバイス名情報の付与に対応し、トンネルテーブルは MN および CN の通信デバイス名フィールドを追加した。これにより、PMM はどの NTMobile-UDP トンネルを使用するかを選択できるようにした。

4.2 拡張 DC のプロトタイプ実装

拡張 DC は、拡張データの付与・取得機能、NTM 端末のアドレス情報を管理する MySQL データベースから MN、CN の複数アドレス情報取得保存機能およびアドレス情報数に応じた複数トンネル構築指示機能を C 言語で実装し

^{*3} <https://developer.android.com/topic/libraries/architecture/room>

^{*4} 送信元 IP アドレス、宛先 IP アドレス、送信元ポート番号、宛先ポート番号、トランスポートプロトコルの組合せ。

た. Node Address Table は 3.3.3 項で述べた dev_name を追加フィールドとしてデータベース構造を拡張した.

5. 評価

提案手法は追加接続先ネットワーク品質確認のためのハンドオーバー検知時間および複数トンネル構築時間が追加されることになるが, ハンドオーバー時におけるトンネル構築処理オーバーヘッドが従来手法よりも大幅に増加せず, 実用上問題ないか確認する. また, 従来の VpnService 型 NTMobile は移動管理機能が実装されていないため, 提案手法単独として接続状態が LTE からマルチホーム, マルチホームから LTE に切り替わった際の通信フローのパケットロスフロー経路点となる RS にて測定し, 提案手法による QoE 低下を防ぐための通信断絶のないシームレス IP Flow Mobility が実現可能か検証する.

5.1 測定環境

図 13 に測定環境を示す. 拡張 DC および RS を AWS EC2 上の東京リージョンに設置し, 各サーバの EC2 インスタンス諸元は表 3 のとおりである. EC2 インスタンスタイプは最低スペックとなる t1.micro であり, 今回は IPv4 のみの通信が利用可能な環境とした. なお, 拡張 DC には拡張 NTMobile と従来の NTMobile の両プログラムを導入し, 提案手法と従来手法における DC として動作を切り替えられるようにした.

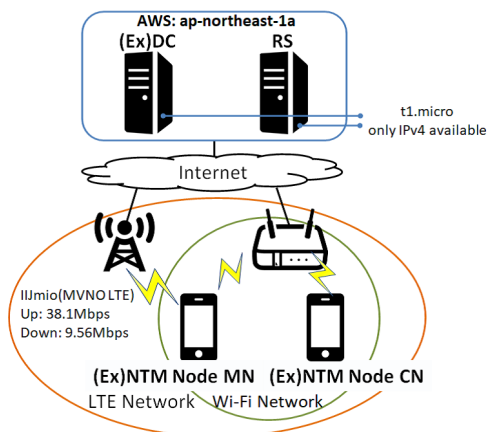


図 13 測定環境

Fig. 13 Measurement environment.

表 3 各サーバの AWS EC2 インスタンス諸元

Table 3 AWS EC2 instance specifications of each server.

	ExDC, RS
Availability Zone	ap-northeast-1a
Instance Type	t1.micro
OS	Ubuntu 14.04 LTS 32 bit
vCPU	1
Memory	0.613 GiB
Networking	Very Low

拡張 NTM 端末には Android スマートフォン (MN: Pixel3, Android 9.0. CN: Pixel3a, Android 9.0) を利用し, 提案手法および従来手法のアプリケーションをそれぞれインストールして, 提案手法と従来手法適用時の検証を実施した. MN は IJmio による MVNO LTE 回線 (docomo) および Wi-Fi5 回線を使用し, CN は MN と同一 NAT 配下の Wi-Fi5 回線を使用した.

また, 測定環境の特性を明確にするために, 各端末間の RTT を測定した. 表 4 に各端末間の RTT を示す. RTT の測定には ping を使用し, 1 秒間隔で 64 バイトのパケットを 100 回送受信し, 測定時刻は平日 16 時 00 分ごろである.

5.2 ハンドオーバー時トンネル構築処理オーバーヘッド

5.2.1 従来手法における測定シナリオ

従来の NTMobile がプロトタイプ実装された VpnService 型 NTMobile アプリケーションでは, 接続ネットワーク変化検出機能が実装されておらず, 完全なハンドオーバー動作を行うことができないが, 従来のハンドオーバー時における DC へのアドレス情報再登録および DC へのトンネル経路指示要求 Direction Request から 1 つの Tunnel Response までの処理時間を測定することは可能である. MN は LTE もしくは Wi-Fi のみに接続し, すでに DC へのアドレス情報初登録を完了しているものとする. CN はつねに Wi-Fi 接続のみであり, すでに DC への Wi-Fi アドレス情報初登録を完了しているものとする. MN が LTE もしくは Wi-Fi を使用して Registration Request 送信から Tunnel Response 受信までの処理時間を各計 10 回測定した. このシナリオにおける具体的手順は以下のとおりである.

- (1) MN は LTE/Wi-Fi 回線を使用して DC に対してアドレス再登録として Registration Request を送信.
- (2) MN は DC から Registration Response を LTE/Wi-Fi 経由で受信し, Direction Request を LTE/Wi-Fi 経由で DC に送信.
- (3) MN は LTE/Wi-Fi 経由で Route Direction を受信し, MN は RS 経由で Tunnel Request を CN に送信.
- (4) MN は RS 経由で CN から LTE/Wi-Fi で Tunnel Response を受信し, トンネル再構築完了.

表 4 各端末間の RTT

Table 4 RTTs between each node.

Section	Min [ms]	Avg. [ms]	Max [ms]
MN _{LTE} – DC	65.95	100.39	383.73
MN _{LTE} – RS	48.16	93.93	271.60
MN _{WIFI} – DC	11.43	30.75	132.43
MN _{WIFI} – RS	12.60	32.85	360.83
CN _{WIFI} – DC	12.11	25.73	119.66
CN _{WIFI} – RS	13.36	24.64	53.31
RS – DC	0.36	0.45	0.76

表 5 LTE ネットワークへのハンドオーバー時における従来手法の各処理時間

Table 5 Processing time of each conventional method during handover to LTE network.

Process	Min [ms]	Avg. [ms]	Max [ms]
1. Address updating	92.91	541.30	1292.29
2. Tunnel re-establishment	304.53	366.95	455.05

表 6 Wi-Fi ネットワークへのハンドオーバー時における従来手法の各処理時間

Table 6 Processing time of each conventional method during handover to Wi-Fi network.

Process	Min [ms]	Avg. [ms]	Max [ms]
1. Address updating	25.48	33.88	96.28
2. Tunnel re-establishment	173.10	224.43	285.44

5.2.2 提案手法における測定シナリオ

MN はマルチホーム状態であり、すでに拡張 DC への LTE アドレス情報登録を完了しているものとする。CN はつねに Wi-Fi 接続のみのシングルホーム状態であり、すでに DC への Wi-Fi アドレス情報登録を完了しているものとする。MN が移動先ネットワークの通信状況確認処理開始から複数 Tunnel Response を受信完了したトンネル追加構築完了までの処理時間を計 10 回測定した。このシナリオにおける具体的手順は以下のとおりである。

- (1) MN が LTE に加えて Wi-Fi に追加接続し、100ms 間隔で Wi-Fi の RSSI および拡張 DC との RTT 値をそれぞれ 3 回取得し、トンネル追加構築実行判断を下す。
- (2) MN が DC に対して拡張 Registration Request を Wi-Fi 経由で送信。
- (3) MN が DC から Registration Response を Wi-Fi 経由で受信。
- (4) MN が Direction Request を Wi-Fi 経由で送信。
- (5) MN が Route Direction を LTE, Wi-Fi 経由で各 1 回受信。
- (6) MN が RS 経由で CN に対して各 1 回 Tunnel Request を LTE, Wi-Fi 経由で送信。
- (7) MN が RS 経由で CN から LTE, Wi-Fi 経由で各 1 回 Tunnel Response を受信してトンネル追加構築完了。

5.2.3 測定結果と考察

従来手法で LTE 回線を使用した MN におけるアドレス情報再登録開始からトンネル再構築までの各処理時間を表 5 に、Wi-Fi 回線使用時の結果を表 6 に示す。提案手法の MN におけるハンドオーバー検知開始から追加トンネル構築までの各処理時間を表 7 に示す。提案手法におけるハンドオーバー検知に関して、1 度取得した RTT 値を考慮して測定間隔を調整し、100ms 以内の RTT 値であればつねに測定間隔が 100ms となるように実装されている。しかし、MN が利用している MVNO LTE 回線では RTT 値が

表 7 ハンドオーバー時における提案手法の各処理時間

Table 7 Processing times of the proposed method during handover.

Process	Min [ms]	Avg. [ms]	Max [ms]
1. Handover detection	485.92	549.55	599.27
2. Address updating	85.12	603.87	957.77
3. Tunnel re-establishment	315.53	400.26	470.05

刻一刻と大幅に上下動するため、1 回目のハンドオーバー判断を下すまでに 300ms 以上の時間がかかる結果となった。接続ネットワークの通信品質を確認するハンドオーバー検知処理時間は従来手法に比べて単純に加算されることになるが、きわめて通信品質の低い移動先ネットワークに切り替えることを防ぐ効果がある。

アドレス登録処理時間に関して、従来手法は LTE では平均 541.30ms、Wi-Fi では平均 33.88ms、提案手法は平均 603.87ms であり、LTE 使用の従来手法および提案手法において最も処理時間がかかるという結果となった。この時間に関して、LTE 使用の従来手法では LTE MVNO による RTT 値の影響を受けたためであると考えられる。提案手法では Wi-Fi を使用して Wi-Fi アドレス情報追加登録を行うが、すでに CN と MN の LTE 側による DC への導通確保のための NTMobile におけるキープアライブが行われており、最低スペックの t1.micro で構成された DC による応答性の低下が要因として考えられる。Wi-Fi 使用の従来手法では DC の処理負荷は比較的少なく、DC との RTT 値に近い値となったと考えられる。従来手法および提案手法はともに最小値と最大値できわめて大きな差が存在し、これは AWS EC2 インスタンスの最低スペックで構築した DC 自体の処理能力が 1 つの要因として考えられる。

トンネル構築処理時間に関して、従来手法は LTE では平均 366.95ms、Wi-Fi では平均 224.43ms、提案手法は平均 400.26ms がかかるという結果となった。提案手法ではマルチホーム通信移行時に LTE と Wi-Fi の両方のトンネルを構築するが、実装として、DC から CN および MN への 1 つ目の拡張 Route Direction が到達後各 NTM 端末は追加のスレッドを立ち上げて、結果として複数の Tunnel Request/Response が並行処理されるため実装上オーバーヘッドは最小となり、単一のトンネルを再構築する LTE 使用の従来手法と比較しても遜色ない結果が得られた。

全体のハンドオーバー時間として、従来手法は LTE では平均 908.25ms、Wi-Fi では平均 258.31ms、提案手法は平均 1,553.68ms となった。これは、RTT 値、DC のスペックにより変動すると考えられ、その要因を除外した場合、提案手法によるハンドオーバー検知処理時間が差の主要因であると考えられる。しかしながら、従来手法ではハンドオーバー時間の間通信断絶が発生するが、提案手法はトンネル追加構築完了後に通信フローが各トンネルに割り当てられ、

移動前の通信が継続されることで通信断絶は発生しない。そのため、ハンドオーバー時間の増加がアプリケーションの通信品質に影響を与えることはなく、QoEの向上あるいは維持を狙うことが可能である。

5.3 接続ネットワーク追加/削除時における通信フロー移動によるパケットロス

5.3.1 測定シナリオ

提案手法を適用したNTM 端末の接続状態がLTE からマルチホーム、マルチホームからLTE に切り替わった際の通信フローのパケットロスを測定する。通信フローとして、VoIP におけるコーデック G.711 を想定し、Android アプリケーション「Magic iperf」*5 を使用して、UDP ペイロードが 160 byte, 50 pps (packets per sec) になるよう設定し、つねに 1,002 個のパケットを生成して送信する。UDP フローに関して Wi-Fi 優先利用として登録し、Wi-Fi ネットワークが利用可能な場合は UDP フローを MN から見て Wi-Fi 側のトンネルに割り当てるよう、MN 側にルーティングポリシーを設定した。

今回は以下に示す 2 つのシナリオを想定し、それぞれ 10 回実験を行う。

- シナリオ 1：シングルホームからマルチホーム
MN が LTE 側トンネル経路で VoIP フローを通信中に Wi-Fi を ON にしてマルチホーム状態となり、トンネル追加構築完了後に VoIP フローを Wi-Fi 側トンネル経路へハンドオーバーする。
- シナリオ 2：マルチホームからシングルホーム
MN が LTE 側、Wi-Fi 側の両方でトンネル構築済みの状態で、Wi-Fi 側トンネル経路で VoIP フローを通信中に MN が移動し、Wi-Fi の切断を通信断絶前に検知して VoIP フローを LTE 側トンネル経路へハンドオーバーする。

なお、図 13 の測定環境においては、MN と CN 間は RS 経由でトンネル通信が行われるため、RS 上で tcpdump を実行し、観測パケットから LTE 側および Wi-Fi 側トンネル経路で受信したパケット数を算出した。また、MN 側においても仮想インタフェースに対して tcpdump でパケット観測し、VoIP フローに関する生成パケット数を確認する。MN 側、RS 側の観測結果を比較することにより、パケットロスの有無を確認した。

5.3.2 結果と考察

図 14、図 15 にそれぞれシナリオ 1、シナリオ 2 における MN 側 VoIP パケット送信数と RS 側 VoIP パケット受信数の比較結果を示す。パケット観測の結果、MN が生成した VoIP フローのパケット数は全 10 回とも 1,002 個で

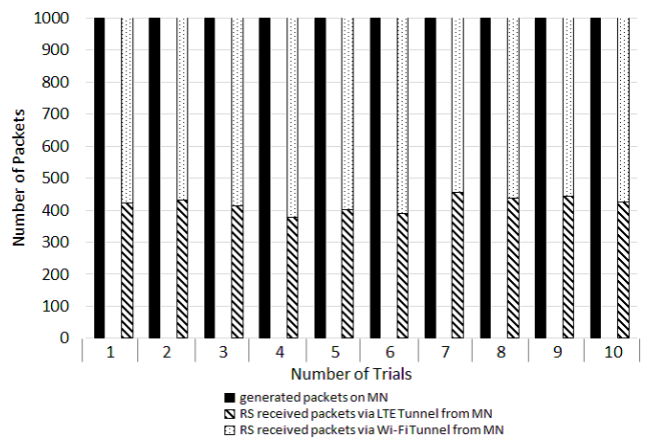


図 14 シナリオ 1 における MN 側送信パケット数と RS 側受信パケット数の比較

Fig. 14 Comparison of the number of packets sent by MN and the number of packets received by RS in Scenario 1.

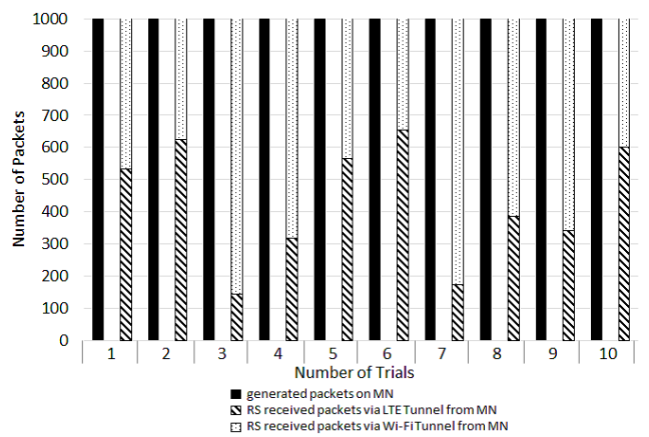


図 15 シナリオ 2 における MN 側送信パケット数と RS 側受信パケット数の比較

Fig. 15 Comparison of the number of packets sent by MN and the number of packets received by RS in Scenario 2.

あった。一方、RS 側で受信した VoIP フローに関するパケット数は、試行ごとにハンドオーバーのタイミングが異なるため LTE 側トンネル経路および Wi-Fi 側トンネル経路で受信したパケット数は異なるものの、両パケットの合計は全 10 回とも 1,002 個であり、フローレベルハンドオーバーによるパケットロスが発生していないことが確認できた。

以上の結果より、提案手法は Android スマートフォンのアプリケーションレベルで通信断絶のない IP Flow Mobility を実現できることを実証できた。

5.4 スループット特性

提案手法では NTM 端末がマルチホーム状態になったとき、パケット送信時にルーティングポリシーを検索する処理が従来手法の VpnService 型 NTMobile に対して追加されているため、スループットが低下する可能性がある。ただし、ルーティングポリシーに記載されるエントリ数は膨大な数とはならないため、検索処理にともなうパケット処理時

*5 <https://play.google.com/store/apps/details?id=com.nextdoordeveloper.miperf.miperf&hl=us&gl=US>

間の大幅な増加は発生しないと考えられる。文献 [22] では従来手法のスループット特性が評価されており、通常の TCP 通信 15.2~15.35 Mbps に対して 13.84~13.95 Mbps と約 9%の減少に抑えられ、カーネルを改造しない方式であってもスループットがあまり低下しない実装が可能であることが報告されている。したがって、提案手法も従来手法と同等な実用上問題のないスループット特性が期待される。

5.5 今後の検討

端末主導型 IP Flow Mobility を実現する既存技術では、開発者が事前に設定した QoS プロファイルを満たすフローをオフロードすることは可能だが、フロー移動に関わる通信断絶については考慮されておらず、またカーネルの改造が必要であり導入が困難であった。これに対して、提案手法はアプリケーションレベルでシームレスな IP Flow Mobility を実証することができ、本論文の主要な目的は達成できた。

提案手法の今後の検討課題として、フロー割当て判断がある。フックした仮想 IP パケットから取得したフロー情報をもとに、既存技術と同様に開発者によって事前に用意された静的なルーティングポリシーを検索することにより、使用する NTMobile トンネルを決定し、フロー割当てがなされる基本機能は確認できた。実運用にあたっては、スマートフォン上で動作する一般のアプリケーションがどのようなポリシーを適用したいのかを拡張 NTMobile アプリケーション側に要求する機能が必要になると考えられる。Android においてはインテントを利用することにより、別の Android アプリケーションにアクションを要求することができる。そのため、あるアプリケーションがインテントを用いてポリシーを拡張 NTMobile アプリケーションに伝えることにより、シームレスハンドオーバーを利用する運用が考えられる。たとえば、インテントにはフロー情報と優先使用したい通信インタフェースを JSON (JavaScript Object Notation) 形式で記載することにより、ルーティングポリシーを設定することができる。

また、NTM 端末が所有するルーティングポリシーに加えて、ネットワーク側のポリシーも考慮することにより、さらに柔軟なフロー割当て決定が可能になると考えられる。3GPP ではモバイルと Wi-Fi の効率的な使用切替えを目的として、LTE などの 3GPP ネットワークと Wi-Fi などの Non-3GPP ネットワークへの接続をポリシー化するルールを 3GPP に準拠した携帯ネットワークから MN に提供する ANDSF (Access network discovery and selection function) が議論されている [23]。ANDSF で提供される代表的な情報として ISRP (Inter-System Routing Policy) があげられ、IP アドレスやアプリケーション名およびドメイン名などで通信を定めて使用するアクセスネットワークを

指定する。拡張 NTMobile 実装端末は端末主導型 IP Flow Mobility であり、端末背景情報を利用できるため、ISRP より通知されたアプリケーション名と使用 IP アドレスを紐づける機能を追加実装することにより、自身のルーティングポリシーを拡張することができると考えられる。

なお、提案手法は VpnService を利用した NTMobile をユーザ空間で拡張しカーネルの改造を不要としているが、Android スマートフォンでマルチホーム通信時に適切な通信インタフェースを利用するために、`iproute2` を用いてルーティングテーブルを使い分ける必要がある。しかし、一般的な Android スマートフォンでは一般ユーザの権限で `iproute2` を利用することができず、現状では root 権限の使用が回避不可能である。今後、Android OS のアップデートによりマルチホーム通信が標準的にサポートされれば、提案手法の動作時に root 権限が不要となり、市販の Android スマートフォンを用いて一般ユーザが容易に提案手法を利用できるようになる。

6. まとめ

本論文では、従来の NTMobile の仕様を拡張することにより、カーネルの改造を不要とした通信断絶の発生しない端末主導型シームレス IP Flow Mobility を実現する手法を提案した。NTMobile サーバと Android スマートフォンに拡張 NTMobile 機能を実装して、実運用されているクラウドおよびモバイルネットワークを利用した動作検証の結果、パケットロスの発生しないシームレス IP Flow Mobility をアプリケーションレベルで実現できることを確認した。また、従来手法との性能比較をした結果、提案手法は従来手法に比べてオーバーヘッドは増加するものの、ハンドオーバー処理にともなう通信断絶は発生しないことから、アプリケーションへの影響は生じないことを確認し、実用上問題ないことを示した。

今後、拡張 NTMobile による IP Flow Mobility のさらなるフロー割当てにおける柔軟性の獲得には、3GPP 準拠携帯ネットワークから提供される ANDSF によるポリシーへの対応が必要になると考えられる。これにより、拡張 NTMobile 側で事前に用意された静的ポリシーとネットワーク側のポリシーの両方を考慮した幅広い通信フローへの対応が可能となり、IP Flow Mobility の有用性が高まると考えられる。

参考文献

- [1] Cisco: Cisco Visual Networking Index: Forecast and Trends, 2017-2022 (2019), available from <https://s3.amazonaws.com/media.mediapost.com/uploads/CiscoForecast.pdf> (accessed 2021-01-14).
- [2] Perkins, C. (Ed.): IP Mobility Support for IPv4, Revised, RFC 5944, IETF (2010).
- [3] Perkins, C. (Ed.): Mobility Support in IPv6, RFC 6275,

- IETF (2011).
- [4] Soliman, H. (Ed.): Mobile IPv6 Support for Dual Stack Hosts and Routers, RFC 5555, IETF (2009).
- [5] Gundavelli, S., Leung, K., Devarapalli, V., Chowdhury, K. and Patil, B.: Proxy Mobile IPv6, RFC 5213 (2008).
- [6] 鈴木秀和, 上醉尾一真, 水谷智大, 西尾拓也, 内藤克浩, 渡邊 晃: NTMobile における通信接続性の確立手法と実装, 情報処理学会論文誌, Vol.54, No.1, pp.367–379 (2013).
- [7] 内藤克浩, 上醉尾一真, 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, 情報処理学会論文誌, Vol.54, No.1, pp.380–393 (2013).
- [8] 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, 情報処理学会論文誌, Vol.54, No.10, pp.2288–2299 (2013).
- [9] IEEE 802.21 Standard: Local and Metropolitan Area Networks – Part 21: Media Independent Handover Services, Technical report, IEEE Computer Society (2009).
- [10] Okubo, Y., Suzuki, H., Naito, K. and Watanabe, A.: A Seamless Handover Method Using IEEE 802.21 and NTMobile for Android Smartphones, *Proc. 9th International Conference on Mobile Computing and Ubiquitous Networking*, Vol.2016, No.12, pp.1–2 (2016).
- [11] 内閣府: 第 5 期科学技術基本計画 (2016).
- [12] Giaretta, G.: IP flow mobility and seamless Wireless Local Area Network (WLAN) offload; Stage 2, TS 23.261, 3GPP, Release 16 (2020).
- [13] de la Oliva, A., Bernardos, C.J., Calderon, M., Melia, T. and Zuniga, J.C.: IP flow mobility: Smart traffic offload for future wireless networks, *IEEE Communications Magazine*, Vol.49, No.10, pp.124–132 (2011).
- [14] Bernardos, C.: Proxy Mobile IPv6 Extensions to Support Flow Mobility, RFC 7864, IETF (2016).
- [15] Melia, T., Bernardos, C.J., de la Oliva, A., Giust, F. and Calderon, M.: IP Flow Mobility in PMIPv6 Based Networks: Solution Design and Experimental Evaluation, *Wireless Personal Communications*, Vol.61, No.4, pp.603–627 (online), DOI: 10.1007/s11277-011-0423-3 (2011).
- [16] Purohith, D.R. and Sivalingam, K.M.: SIFM: A network architecture for seamless flow mobility between LTE and WiFi networks – Analysis and Testbed Implementation, *CoRR*, Vol.abs/1702.07489 (2017).
- [17] Varga, N., Bokor, L., Bouroz, S., Lecroart, B. and Takács, A.: Client-based and Cross-layer Optimized Flow Mobility for Android Devices in Heterogeneous Femtocell/Wi-Fi Networks*, *Procedia Computer Science*, Vol.40, pp.26–36 (online), available from <https://doi.org/10.1016/j.procs.2014.10.028> (2014). Fourth International Conference on Selected Topics in Mobile & Wireless Networking (MoWNet' 2014).
- [18] Wakikawa, R., Devarapalli, V., Tsirtsis, G., Ernst, T. and Nagami, K.: Multiple Care-of Addresses Registration, RFC 5648, IETF (2009).
- [19] Tsirtsis, G., Giarreta, G., Soliman, H. and Montavont, N.: Traffic Selectors for Flow Bindings, RFC 6088, IETF (2011).
- [20] Tsirtsis, G., Soliman, H., Montavont, N., Giaretta, G. and Kuladinithi, K.: Flow Bindings in Mobile IPv6 and Network Mobility (NEMO) Basic Support, RFC 6089, IETF (2011).
- [21] Koodli, R. (Ed.): Mobile IPv6 Fast Handovers, RFC 5568, IETF (2009).
- [22] Yamada, T., Suzuki, H., Naito, K. and Watanabe, A.:

IP Mobility Protocol Implementation Method Using VpnService for Android Device, *Proceedings of The Ninth International Conference on Mobile Computing and Ubiquitous Networking*, Vol.2016, No.16, pp.1–2 (2016).

- [23] Sedlacek, I.: Access Network Discovery and Selection Function (ANDSF) Management Object (MO), TS 24.312, 3GPP, Release 16 (2020).



松岡 穂 (学生会員)

2019 年名城大学理工学部情報工学科卒業。2021 年同大学大学院理工学研究科情報工学専攻修士課程修了。2021 年 SCSK 株式会社に入社。在学時代は主としてモバイルネットワークにおけるフローレベルハンドオーバーに関する研究に従事。修士 (工学)。



鈴木 秀和 (正会員)

2004 年名城大学理工学部情報科学科卒業。2009 年同大学大学院理工学研究科電気電子・情報・材料工学専攻博士後期課程修了。2008 年日本学術振興会特別研究員。2010 年名城大学理工学部助教。2015 年同大学理工学部准教授および東北大学電気通信研究所共同研究員を兼務。2020 年名古屋大学未来社会創造機構モビリティ社会研究所特任准教授を兼職。モバイルネットワークおよびユビキタスコンピューティング等の研究に従事。博士 (工学)。IEEE, ACM, 電子情報通信学会各会員。本会シニア会員。



内藤 克浩 (正会員)

1999 年慶應義塾大学理工学部電気工学科卒業。2004 年名古屋大学大学院工学研究科情報工学専攻博士課程後期課程修了。2004 年三重大学工学部電気電子工学科助手。2007 年同大学助教。2011 年カリフォルニア大学ロサンゼルス校客員研究員。2014 年愛知工業大学情報科学部准教授。2016 年情報処理学会・長尾真記念特別賞受賞。無線ネットワーク, モバイルコンピューティングの研究に従事。博士 (工学)。IEEE, 電子情報通信学会各シニア会員。本会シニア会員。