

IEEE 802.21 を利用した Android スマートフォンの ハンドオーバに関する基礎的実験

大久保 陽平^{1,a)} 鈴木 秀和^{1,b)} 内藤 克浩² 渡邊 晃¹

概要: 仮想 IP アドレスとトンネリング技術により移動透過性を実現する技術として, NTMobile (Network Traversal with Mobility) が提案されている. NTMobile では, NTMobile 搭載端末がルーティングテーブルの変化により, 無線インタフェースの切り替えを検知することでハンドオーバ処理を実行している. しかしこの手法は, ハンドオーバ処理の実行が無線インタフェースの切り替え後となるため, ハンドオーバに伴う通信断絶時間が発生してしまうという課題がある. 本稿では, NTMobile をハンドオーバの支援技術である IEEE 802.21 と連携させることでシームレスハンドオーバを実現する手法を提案する. この手法は, IEEE 802.21 の機能を利用し, 端末の接続するネットワークが切り替わる前に事前にハンドオーバ処理を行うことで, 通信断絶時間を削減するものである. 提案手法の一部を実装して基礎的な動作検証実験を行った結果, 通信中にネットワークが切り替わっても通信が継続され, IEEE 802.21 と NTMobile の連携機能が正しく動作していることを確認した.

1. はじめに

スマートフォンやタブレット端末の普及により, Wi-Fi のアクセスポイントを利用できる場合は Wi-Fi に接続し, それ以外では LTE に接続するなど, Wi-Fi や LTE などの異なるネットワークを切り替えて通信する機会が増加している. しかしインターネットで主に利用されている TCP/IP ネットワークでは, 端末に割り当てられた IP アドレスとポート番号を用いて通信を行っているため, 例えば端末が LTE のネットワークから Wi-Fi のネットワークへ接続を切り替えた場合, ネットワークの切り替えにより IP アドレスに変化が生じ, 移動前に行っていた通信が移動後に継続できないという課題がある. そこで, このような課題を解決できる技術である移動透過性技術が必要とされており, これまでに Mobile IPv4 [1] や Mobile IPv6 [2] など様々な移動透過性技術が提案されてきた. しかし Mobile IPv4 では, 移動透過性を実現するために常に HA (Home Agent) を経由するため, 通信経路が冗長になってしまう課題があり, また Mobile IPv6 では, 現在でもネットワーク

の主流である IPv4 環境に適応できないなど, 実用上の課題があった. そこで筆者らは, IPv4/IPv6 混在環境において移動透過性を実現できる技術である NTMobile (Network Traversal with Mobility) [3,4] を提案している.

NTMobile は Linux OS での実装を完了しており, Linux ベースである Android 端末へ実装することができる. Android 端末の多くが Wi-Fi の無線インタフェースと 3G や LTE などの携帯電話網の無線インタフェースを搭載しており, Android OS を搭載した NTMobile 搭載端末 (NTM 端末) では, Wi-Fi と携帯電話網を切り替えて通信を行うケースが多数発生することが予想される. しかし従来の NTMobile は, NTM 端末がハンドオーバする際に発生する移動先ネットワークからの実 IP アドレス取得処理や NTM 端末間のトンネル再構築処理に伴って数秒の通信断絶時間が発生してしまう. そのためリアルタイム性が要求されるアプリケーションの利用を考慮して, この通信断絶時間が課題となっている.

本稿では, NTMobile の実用化における課題である NTM 端末がハンドオーバする際に発生する通信断絶時間を削減し, シームレスなハンドオーバを実現する手法を提案する. ハンドオーバ支援技術であり, ネットワーク切り替え前に移動先のネットワークのハンドオーバ処理の実行を可能とする技術である IEEE 802.21 (MIH:Media Independent Handover) [5] と NTMobile を連携させることにより, シームレスハンドオーバを実現する. 提案手法では, IEEE

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
Department of Information Science, Aichi Institute of Technology

a) youhei.okubo@ucl.meijo-u.ac.jp

b) hsuzuki@meijo-u.ac.jp

802.21 の機能を利用して定期的に通信中のインタフェースの電波強度を測定し、電波強度の劣化を検出するとハンドオーバー処理を開始する。ハンドオーバー処理では、IEEE 802.21 の機能を利用して切り替え先の無線インタフェースをネットワークに接続させ、その後に NTMobile による NTM 端末間のトンネル再構築処理を行う。ハンドオーバー処理を行っている間は、切り替え前の無線インタフェースを用いて通信を維持し、ハンドオーバー処理が完了した後に通信を行う無線インタフェースを切り替えることでシームレスハンドオーバーを実現する。

以下、2 章で NTMobile の概要、4 章で提案手法の動作について述べる。また 5 章で実装と提案手法の評価について述べ、最後に 6 章でまとめる。

2. NTMobile

2.1 システム構成

NTMobile のシステム構成を図 1 に示す。NTMobile は、NTM 端末に対して端末の位置に依存しない通信識別子である仮想 IP アドレスの割り当てや NTM 端末間の通信に利用される UDP トンネルの構築指示を行う機能を持つ DC (Direction Coordinator), 通信を行う端末の一方が IPv4 ネットワークに接続し、もう一方が IPv6 ネットワークに接続する場合など、直接通信ができない場合に通信の中継を行う機能を持つ RS (Relay Server) と NTMobile を実装した端末である NTM 端末で構成される。また DC および RS は IPv4/IPv6 ネットワーク混在環境であるデュアルスタックネットワークに接続し、ネットワーク上に複数台設置することが可能である。

2.2 ハンドオーバー処理

Wi-Fi ネットワークに接続された NTM 端末 MN が LTE ネットワークにハンドオーバーする際の処理を図 2 に示す。また LTE ネットワークから Wi-Fi ネットワークにハンドオーバーする際の処理の概要を図 3 に示す。NTM 端末におけるハンドオーバー時の処理は、通信インタフェース切り替えなど L2 (レイヤ 2) のハンドオーバー処理、基地局や AP (アクセスポイント) へのネットワーク接続処理、実 IP アドレスの取得処理、DC へのアドレス情報の登録処理と MN と CN 間でのトンネル再構築処理からなる。以下、これらの処理の詳細について述べる。

2.2.1 ネットワーク接続処理

図 2 および図 3 のように、接続するネットワークが切り替わった場合、移動先のネットワークへの接続処理が発生する。このとき MN は携帯電話網の基地局や Wi-Fi の AP へ接続処理を行い、その後、IPv4 プライベートネットワークであれば DHCP (Dynamic Host Configuration Protocol) により移動先のネットワークから実 IPv4 アドレスを取得する。接続先のネットワークが IPv6 ネットワークである

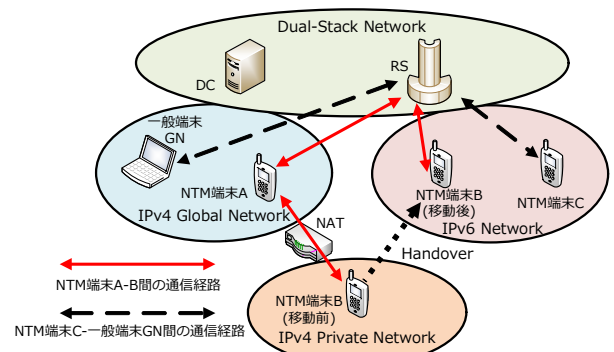


図 1 NTMobile の概要

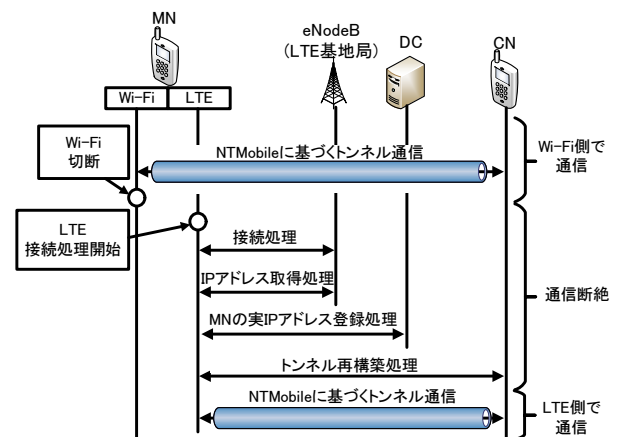


図 2 Wi-Fi から LTE へのハンドオーバー時の処理

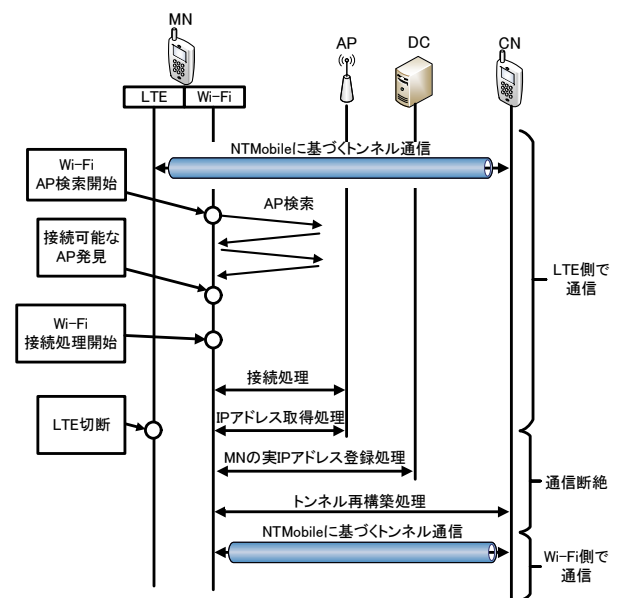


図 3 LTE から Wi-Fi へのハンドオーバー時の処理

場合、ネットワークプレフィックスと IPv6 ネットワーク内において一意となる値であるインタフェース ID により IPv6 アドレスを自動生成し、端末に設定する。端末の実 IP アドレスは接続先のネットワークから割り当てられるため、端末の接続するネットワークの切り替えに伴い、端末

の実 IP アドレスが変化する。なお DC から割り当てられた仮想 IP アドレスは、MN の接続するネットワークが切り替わっても変化しない。

2.2.2 アドレス登録処理

NTMobile では、NTM 端末に搭載される NTMobile の機能を実行するプログラムである NTM デーモンが常に端末のルーティングテーブルを監視する。NTM デーモンがネットワークへの接続や移動に伴うルーティングテーブルの変化を検知すると、NTMobile のハンドオーバー処理を開始する。NTM 端末は、端末起動時や接続するネットワークが切り替わり、移動先のネットワークから実 IP アドレスを取得すると、自身のアドレス情報を DC に登録する処理である Registration Request を実行する。NTM 端末が常に最新のアドレス情報を自身を管理する DC に登録することで、NTM 端末に対する通信到達性を確保する。

2.2.3 トンネル再構築処理

図 2 および図 3 のように、MN の接続するネットワークが切り替わり、IPv4 プライベートネットワークへ接続された場合の動作を図 4 に示す。MN は接続するネットワークが切り替わると、自身を管理する DC である DC_{MN} へ Registration Request を送信し、実 IP アドレス情報の登録を行う。DC_{MN} は、登録された MN のアドレス情報と CN のアドレス情報に応じて最適なトンネル通信経路を指示し、トンネルを再構築する。MN と CN 間でトンネルを再構築した後、NTMobile に基づく通信を再開する。なお MN と CN で動作しているアプリケーションの通信はネットワークが移動しても変化しない仮想 IP アドレスに基づいて行われているため、MN の接続するネットワークの移動に伴い実 IP アドレスが変化しても、通信の継続が可能である。

2.3 検討課題

2.3.1 NTMobile における課題

従来の NTMobile は、NTMobile の機能を実行するプログラムである NTM デーモンが、NTM 端末のルーティングテーブルを監視し、ルーティングテーブルの変化を検知することで、DC への実アドレスの登録処理とトンネル再構築処理を実行していた。しかしこの手法では、ネットワーク接続処理が完了した後でないとネットワークの移動を検知できず、ハンドオーバーに伴う通信断絶時間が発生してしまうという課題がある。

2.3.2 Android における通信インタフェース切り替え動作の課題

図 5 に、Android 端末における無線インタフェース切り替え時の動作を示す。Android 端末では、携帯電話網から Wi-Fi にハンドオーバーする場合、携帯電話網で接続している間に Wi-Fi から実 IP アドレスを取得する。その後、端末は携帯電話網の無線インタフェースをダウンさせ、Wi-Fi

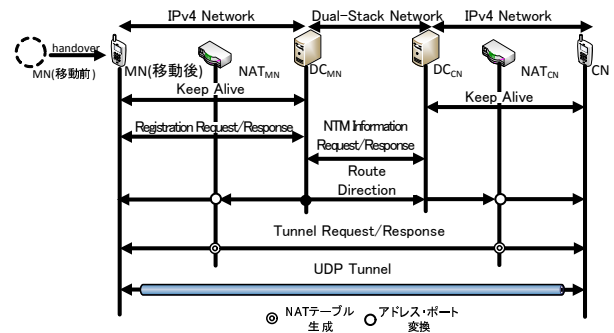


図 4 ハンドオーバー時のトンネル構築シーケンス

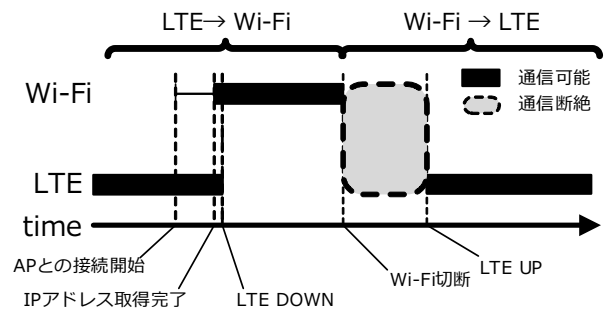


図 5 Android OS におけるハンドオーバー時の動作

側から通信を開始する。そのため NTM 端末では、端末が Wi-Fi へハンドオーバーする場合は、実 IP アドレス取得後のトンネル再構築処理およびアドレス登録処理の時間だけ通信断絶時間が発生する。

一方で Wi-Fi から携帯電話網へハンドオーバーする場合、Android 端末では、Wi-Fi 接続時は携帯電話網の無線インタフェースをダウンしてしまうため、Wi-Fi 接続中に携帯電話網から実 IP アドレスを取得することができない。そのため NTM 端末では、端末が携帯電話網へハンドオーバーする場合は、実 IP アドレスの取得、トンネル再構築処理およびアドレス登録処理の時間だけ通信断絶時間が発生する。また携帯電話網からの実 IP アドレスの取得には数秒程度必要であることが判明しており [4]、Wi-Fi から携帯電話網へのハンドオーバーでは、多くのパケットロスが発生してしまうことが予想される。よって NTMobile により移動透過性が実現できても、ハンドオーバーに伴う通信断絶時間が発生してしまうため実用的とはいえず、解決しなければならない課題の一つとなっている。

3. 関連研究

3.1 Fast Handover Mobile IPv6

FMIP(Fast Handover Mobile IPv6) [6] は、Mobile IPv6 を拡張し高速ハンドオーバーの機能を追加したものである。Mobile IPv6 のハンドオーバー処理は、AP の切断検出、通信リンク確立、移動先ネットワークから割り当てられる IP アドレスである CoA (Care of Address) の取得と HA

と通信相手 CN への移動完了通知からなる。FMIP では、端末は AP から切断される前に近隣の接続可能な別の AP を検索し、移動先で利用する CoA を取得する。これにより、ハンドオーバー時の CoA の取得にかかる通信断絶時間を削減する。また Mobile IPv6 では、移動後の CoA である NCoA (New Care of Address) を HA に登録して利用できるようになるまで通信が断絶してしまうが、FMIP では、その間は移動前の CoA である PCoA (Previous Care of Address) を利用して通信を行うことで、シームレスハンドオーバーを実現する。

しかし FMIP の実現には AP 同士の連携処理が必要であり、端末と AP の両方が FMIP に対応していなければならない。既存の AP は基本的には FMIP に対応できず、全ての機器を FMIP に対応させることは難しい。さらに FMIP では、LTE や Wi-Fi など異なる無線ネットワーク間のシームレスハンドオーバーが考慮されていないという課題がある。

3.2 MIMO を活用したシームレスハンドオーバー手法

スマートフォンや AP の多くは IEEE 802.11n に対応しており、無線インタフェース内に複数のアンテナを搭載し、それぞれのアンテナが通信を行うことで伝送速度を向上させる MIMO (Multiple Input Multiple Output) が使われている。IEEE802.11n において、本来は伝送速度を向上させるためのアンテナのうち、ハンドオーバー時は一つを通信継続用として利用し、もう一つをハンドオーバー用として利用することで Wi-Fi の AP 間におけるシームレスハンドオーバーを実現する手法 [7] が提案されている。

この手法では、通信中のネットワークの通信品質が劣化したら、IEEE 802.11n の省電力モードを用いることにより、通信用アンテナを一本にすることで、ハンドオーバー用のアンテナを確保する。これにより、通信用のアンテナで通信を継続しながら、ハンドオーバー用のアンテナで他の接続可能な AP を検索し、ハンドオーバー処理を行うことができる。ネットワーク側の装置に改造を加えることがなく Wi-Fi の AP 間のシームレスハンドオーバーを実現できることから、Wi-Fi の AP 間のハンドオーバーであれば有用な手法であるといえる。しかし Wi-Fi と携帯電話網間のシームレスハンドオーバーが考慮されておらず、スマートフォンでの実用には課題がある。

3.3 IEEE 802.21

IEEE 802.21 (MIH:Media Independent Handover) は、IEEE によって 2008 年に標準化された規格である。IEEE 802.21 は、L3 (レイヤ 3) など上位レイヤのプログラムが無線インタフェースの規格によらず、統一的に無線インタフェースを扱えるようにすることで、異種無線ネットワーク間のシームレスハンドオーバーを実現する技術である。IEEE 802.21 は、異種無線ネットワーク間のハンドオーバー

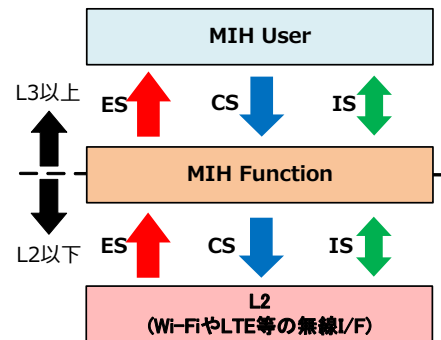


図 6 IEEE 802.21 のフレームワーク

における課題である無線インタフェースごとの仕様の違いを吸収する機能を持つ。IEEE 802.21 の導入により、上位のレイヤは、複数の無線インタフェースを統一的に扱うことが可能になる。

図 6 に IEEE 802.21 のフレームワークを示す。IEEE 802.21 は、MIH User, MIH Function, SAP の 3 つの要素から構成される。MIH User は、L3 に位置する機能であり、異種無線ネットワーク間の移動管理を行う機能を持つ。MIH Function は、L2 と L3 の間に新たに定義されるレイヤであり、無線インタフェースごとの仕様の違いを吸収する機能を持つ。また Link_SAP は、MIH_SAP と MIH_Link_SAP に分けられ、上位または下位のレイヤと MIH Function 間でメッセージの伝達を行う機能を持つ。

IEEE 802.21 では、MIH User が複数の無線インタフェースを統一的に扱うために ES (Event Service), CS (Command Service), IS (Information Service) の 3 つのサービスを定義している。ES は、リンクの確立や切断などリンクの状態に関する情報を扱い、L2 から MIH User へ送信されるサービスである。CS は、電波強度の測定要求、リンクの確立要求やリンクの切断要求などハンドオーバーに関する情報を扱い、MIH User から L2 へ送信されるサービスである。また IS は、ネットワークの最大伝送レートなどの静的な情報を扱い、MIH User と L2 の間で相互に送受信されるサービスである。IS で伝達される情報は、ハンドオーバー先のネットワークを選択する際に用いられる。これらの統一されたサービスを用いることにより、MIH User は、複数の仕様の異なる無線インタフェースを統一的に扱うことが可能となる。

文献 [8] では、IEEE 802.21 と IPv6 ネットワークにおいてネットワーク単位の移動透過性を実現する技術である NEMO (Network Mobility) [9] の連携によるシームレスハンドオーバー手法の提案がされており、実機による動作検証が行われている。動作検証ではマルチホーム環境の構築のために PC が用いられているが、本稿でターゲットとしている Android 端末では 2.3.2 項に述べた課題のために、PC のようにマルチホーム環境を構築することはできない。

4. 提案手法

4.1 概要

NTM 端末は異なる無線ネットワークへハンドオーバーする際に通信断絶時間が発生してしまうという課題があった。さらに NTM 端末が Android スマートフォンの場合は、通信インタフェースの切り替えにおける課題があった。そこで Android OS のアプリケーションフレーム層内に存在するサービスに改造を加えることで通信インタフェースの切り替え動作の課題を解決し、IEEE 802.21 と NTM Mobile を連携させることにより、端末側の実装で Wi-Fi と携帯電話網といった異なる無線ネットワーク間でシームレスハンドオーバーを実現する手法を提案する。

4.2 IEEE 802.21 と連携した NTM Mobile のハンドオーバー時の動作

従来の NTM Mobile はルーティングテーブルの変化により、ネットワークの移動を検知してハンドオーバー処理を実行していたため、通信断絶時間が発生していた。提案手法では、シームレスハンドオーバーを実現するために、NTM デーモンと MIH User に連携機能を追加する。IEEE 802.21 の機能を利用して無線通信の品質を監視し、無線インタフェースを切り替える前に移動先のネットワークのハンドオーバー処理と NTM Mobile のトンネル再構築処理を行うように変更することで、シームレスハンドオーバーを実現する。提案手法の動作の概要を図 7 に示す。また提案手法の動作シーケンスを図 8 に示す。

無線インタフェースの管理には、ES に定義されているリンク確立を示すメッセージである MIH.Link.Up とリンク切断を示すメッセージである MIH.Link.Down、さらに CS に定義されているリンク確立命令やリンク切断命令を示すメッセージである MIH.Link.Action を用いる。またハンドオーバー処理の実行判断には、CS に定義されている電波強度の測定命令を示すメッセージである MIH.Link.Get.Parameters と ES に定義されている電波強度の測定結果を示すメッセージである MIH.Link.Parameters.Report を用いる。

提案手法では、接続しているネットワークの電波強度をハンドオーバーのトリガとして用いる。MIH User は、接続しているネットワークの電波強度測定のために、定期的に通信中の無線インタフェースへ MIH.Link.Get.Parameters を送信する。そして、無線インタフェースから送信される電波強度の測定結果を記載した MIH.Link.Parameters.Report を MIH User が受信することにより、MIH User は、接続しているネットワークの電波強度を得る。電波強度の測定を行った結果、電波強度が閾値以下であった場合、異なる無線ネットワーク間のハンドオーバー処理を開始する。ハンドオーバー処理では MIH User が、端末に搭載されている通信を

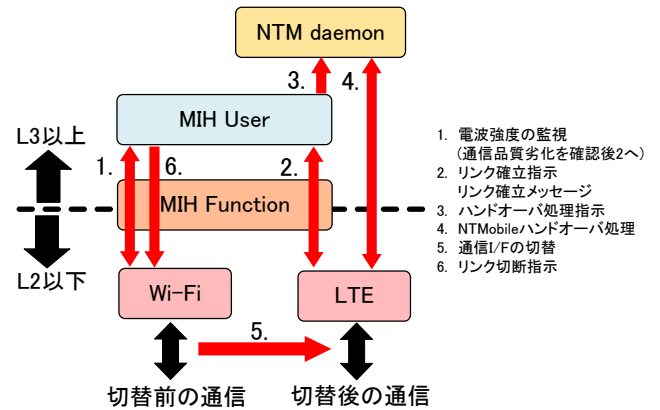


図 7 提案手法のハンドオーバー処理の概要

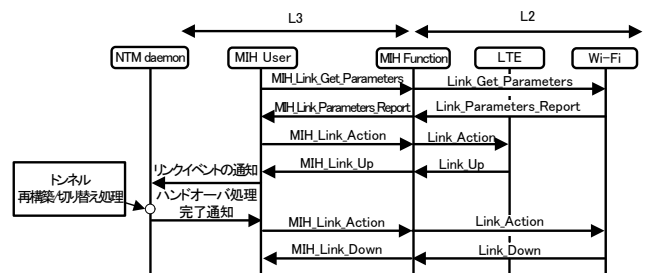


図 8 提案手法の動作シーケンス

行っていない無線インタフェースへ MIH.Link.Action を送信する。MIH.Link.Action を受信した無線インタフェースは、リンク確立を行い MIH User へ MIH.Link.Up を送信する。これにより MIH User は、無線インタフェースのリンク確立を確認する。

MIH User がリンク確立を確認した後、MIH User は、NTM デーモンへリンクイベントの通知により NTM Mobile のハンドオーバー処理の実行を指示する。NTM デーモンは、このメッセージをトリガに新しくリンクを確立した無線インタフェース側からハンドオーバー処理を行う。これにより、NTM Mobile に基づく通信は、新しく接続されたネットワーク側に切り替わって継続される。ハンドオーバー処理の完了報告を受けた MIH User は、ハンドオーバー前に使用していた無線インタフェースに対して MIH.Link.Action を送信することで、リンク切断指示を行い、ネットワークから切断させる。

提案手法を導入した NTM 端末 MN が、Wi-Fi ネットワークから LTE ネットワークにハンドオーバーした場合の処理を図 9 に示す。また LTE ネットワークから Wi-Fi ネットワークにハンドオーバーした場合の処理を図 10 に示す。図 2 と図 3 に示した従来手法と異なり、提案手法では、IEEE 802.21 の機能を利用して、通信断絶前に事前にハンドオーバー処理を行う。ハンドオーバー処理が終了するまでハンドオーバー処理開始前に使用していた無線インタフェースで通信を行うことで通信断絶時間は理論上発生せず、シームレスハンドオーバーを実現できる。

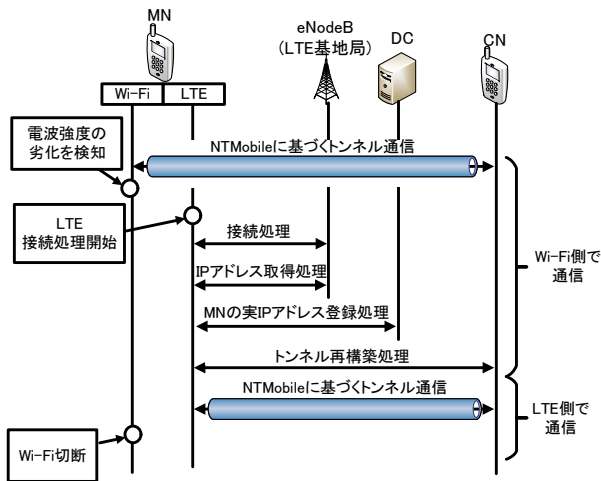


図 9 提案手法の Wi-Fi から LTE へのハンドオーバー時の処理

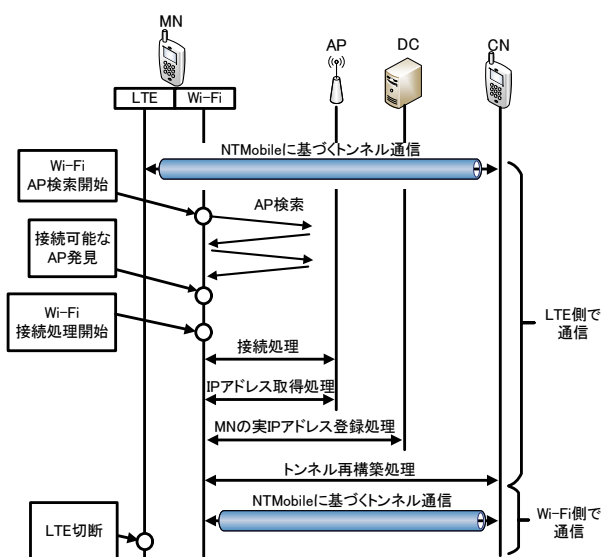


図 10 提案手法の LTE から Wi-Fi へのハンドオーバー時の処理

4.3 Android 端末における通信インタフェース切り替え動作の改善

2.3.2 節で挙げた通信インタフェースの切り替え動作の課題については、Android OS のアプリケーションフレーム層内に存在するサービスである端末の通信管理を行う Connectivity Service に改造を加えることで解決することができる [10]。Connectivity Service を改造することで、通常の Android OS と異なり、Wi-Fi の実 IP アドレスを取得しても携帯電話網の無線インタフェースをダウンさせないようにし、Android スマートフォンにおいてマルチホーム環境を実現することができる。

5. 実装と評価

5.1 実装

IEEE 802.21 の実装として、IEEE 802.21 のオープンソース実装である ODTONE (Open Dot Twenty One) [11] を利用した。ODTONE は、Linux OS、Android OS と Windows で動作が確認されている。ODTONE と NTM デー

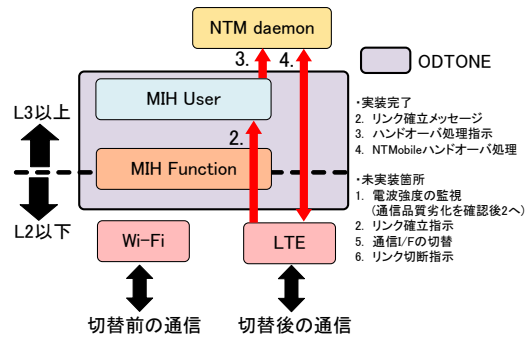


図 11 基礎的実装による動作

モンのプログラム変更を加え、提案手法の基礎的実装を行った。基礎的実装では、NTM 端末を異なる無線ネットワーク間でハンドオーバーさせ、ODTONE と NTMobile の連携機能が正常に動作しているか確認できるように実装を行った。基礎的実装における提案手法の動作を図 11 に示す。

IEEE 802.21 のリンクイベントは、通常であれば MIH Function を経由して MIH User や Link.SAP へ伝達される。しかしデフォルトの ODTONE は、端末を機内モードに切り替えるなど完全に無線インタフェースが切断されると MIH Function から MIH User へ MIH.Link_Down メッセージが送信されず、また端末を機内モードから復帰させた場合でないと、MIH Function から MIH User へ MIH.Link_Up メッセージが送信されないという不具合があった。機内モード時のみ動作することは実用的な動作ではないため、電波が届かなくなり Wi-Fi ネットワークから切断される場合や Wi-Fi の AP を発見して Wi-Fi ネットワークに接続する場合など、ネットワークが自動的に切り替わった際に MIH.Link_Up や MIH.Link_Down が送信されるように ODTONE の MIH Function のプログラムに修正を加えた。さらに ODTONE の MIH User に、MIH Function から MIH.Link_Up を受信すると、UNIX ドメインソケットを生成し、NTM デーモンへハンドオーバー指示メッセージを送信するよう拡張を加えた。

NTM デーモンには、MIH User から伝達されるハンドオーバー指示メッセージの受信機能を UNIX ドメインソケットを利用して追加した。さらに、そのメッセージをトリガに DC へ自身の実 IP アドレス情報の登録処理と端末間のトンネル再構築処理を行うよう変更した。

シームレスハンドオーバーを実現するためには、Android OS の改造と提案手法の全ての実装が必要である。しかし現行の Android OS に対して Connectivity Service の改造が完了しておらず、Android OS における通信インタフェース切り替え時の課題を解決できていない。また ODTONE の通信品質の変化に応じてハンドオーバー指示を行う部分の実装が完了しておらず、完全にシームレスハンドオーバーを実現することができていない。今後は、現行の Android OS に対して改造を行い、また通信品質の監視を行うよう ODTONE の実装を進める必要がある。

5.2 評価

5.2.1 概要

提案手法の基礎的実装を行い、ハンドオーバーに関する実験を行った。実験では IEEE 802.21 と NTMobile の連携が正しく動作するか検証した。さらに端末が Wi-Fi から LTE にハンドオーバーする場合と、端末が LTE から Wi-Fi にハンドオーバーする場合の通信断絶時間の測定を行った。

5.2.2 測定環境

図 12 および表 1 に検証環境の構成と機器仕様を示す。DC と RS は、それぞれ IPv4/IPv6 グローバルネットワークに接続した。MN と CN は、Android 4.4.2 を搭載した Nexus 5 を利用した。またそれぞれを IEEE 802.11n により IPv4 プライベートネットワークに接続させ、さらに MN は IPv4 プライベートネットワークの LTE ネットワークにも接続できるように設定した。MN は Wi-Fi から切断された際に、自動的に LTE ネットワークへ接続処理を行うよう設定した。

5.2.3 評価方法

提案手法の基礎的実装を行った ODTONE と NTM デモンを 2 台の Android スマートフォン Nexus 5 (MN と CN) に導入した。MN から CN への Ping 実行中に MN を手動で Wi-Fi と携帯電話網間をハンドオーバーさせ、ODTONE と NTMobile の連携機能が正常に動作するか確認を行った。通信断絶時間の測定には、CN に導入した tcpdump を使用してパケットをキャプチャし、Wireshark を使用して解析した。ここで、MN 側でなく CN 側でパケットキャプチャを行っている理由は、2.3.2 節で述べたように、Android 端末では、Wi-Fi 接続時に携帯電話網の無線インタフェースが強制的にダウンしてしまうため、MN が携帯電話網にハンドオーバーした際のパケットを MN 側でキャプチャすることができないためである。そのため MN がハンドオーバーする直前に送信したパケットを CN が受信した時刻と、MN と CN 間でトンネルを再構築し、通信が再開されるまでの時刻の差分からハンドオーバー時の全体の通信断絶時間を明らかにした。

本実験では、通信断絶時間の測定を 15 回行いその平均を求めた。また通信断絶時間を MN がハンドオーバーしてからトンネルが再構築されるまでの時間と、トンネルが再構築されてから通信が再開されるまでに分け、内訳を明らかにした。さらに提案手法を導入していない従来の NTMobile を導入した端末で同様の実験を行い、通信断絶時間の測定と比較を行った。

5.2.4 測定結果と考察

MN を Wi-Fi から LTE へハンドオーバーさせた際の通信断絶時間の 15 回の平均値を図 13 に示す。また MN を LTE から Wi-Fi へハンドオーバーさせた際の通信断絶時間の 15 回の平均値を図 14 に示す。また Wi-Fi から LTE へハンドオーバーした場合の最大値と最小値を図 15 に、LTE から

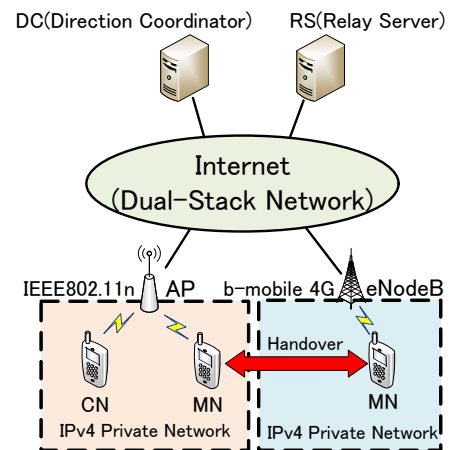


図 12 測定環境

表 1 装置仕様

	DC, RS	MN, CN
Hardware	Dell PowerEdge R415	LG Nexus 5
OS(Kernel)	Ubuntu 10.04(Linux 2.6.32)	Android 4.4.2(Linux 3.4.0)
CPU	AMD Opteron Processor 4180 2.6GHz(仮想 1 コア)	Qualcomm Snapdragon 800 MSM8974 2.26GHz
Memory	512MB	2GB

Wi-Fi へハンドオーバーした際の通信断絶時間の最大値と最小値を図 16 に示す。

測定の結果から、提案手法の基礎的実装の段階では、通信断絶時間が発生してしまっていることが分かった。端末が Wi-Fi から LTE へハンドオーバーした場合の通信断絶時間の平均値は、提案手法の基礎的実装が 2.37 秒、従来の NTMobile が 2.44 秒であり大きな差は発生しなかった。LTE から Wi-Fi へハンドオーバーした場合の通信断絶時間の平均値は、提案手法の基礎的実装が 0.73 秒、従来の NTMobile が 0.51 秒であり、提案手法の基礎的実装の方が通信断絶時間が長いという結果となった。またそれぞれの分散は、端末が Wi-Fi から LTE へハンドオーバーした場合、提案手法の基礎的実装が 1.78、従来の NTMobile が 1.93 であった。LTE から Wi-Fi へハンドオーバーした場合は、提案手法の基礎的実装が 0.09、従来の NTMobile が 0.08 であった。図 13 から図 16 までに示した結果を見ると、通信断絶時間の多くがトンネル再構築が完了するまでの時間が占めていることが分かる。この時間については、過去に行った実験 [12] により、ネットワークへの接続処理が大半を占めていることが確認されている。また通信断絶時間はばらつきが大きく、今後ハンドオーバーのトリガとなる電波強度を決定する際に、通信断絶時間のばらつきを考慮する必要があると考えられる。

今回の実験を行うことで、今後 2.3.2 項に示した Android OS の通信インタフェース切り替えの課題の解決と 5.1 節に示した今回未実装となっている ODTONE の処理の実装を行う必要がある、またハンドオーバーのトリガとなる電波強度を適切に定め、再度実験と評価を行う必要があることが分かった。

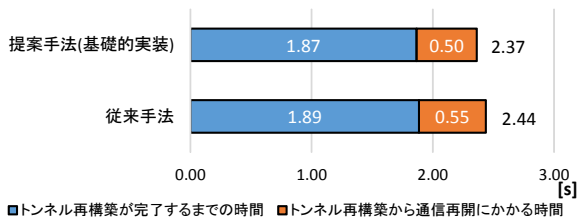


図 13 通信断絶時間の測定結果 (Wi-Fi から LTE)

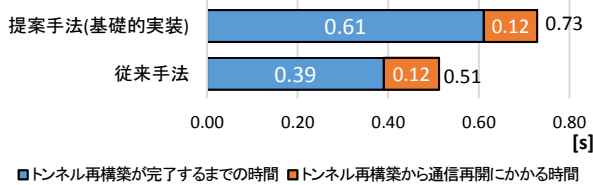


図 14 通信断絶時間の測定結果 (LTE から Wi-Fi)

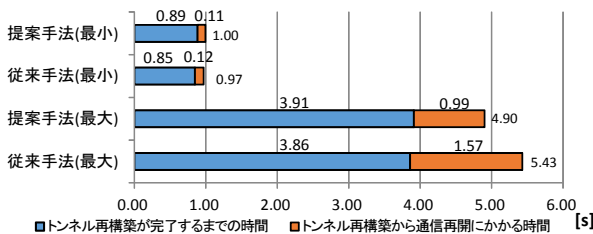


図 15 通信断絶時間の最大値と最小値 (Wi-Fi から LTE)

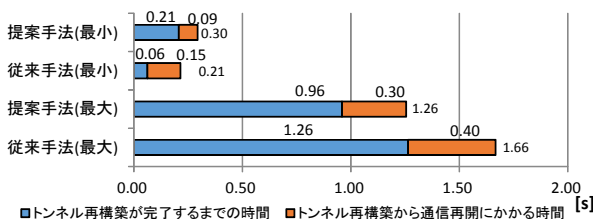


図 16 通信断絶時間の最大値と最小値 (LTE から Wi-Fi)

6. まとめ

本稿では、IEEE 802.21 と NTMobile を連携させることで従来の NTMobile の課題であった通信断絶時間を削減し、シームレスハンドオーバーを実現する手法を提案した。提案手法の基礎的実装を行い、実機で動作検証を行うことで、IEEE 802.21 と NTMobile の連携が正常に動作していることを示した。また今回の実験を行うことで、今後は 2.3.2 項に示した Android OS における通信インタフェース切り替えの課題の解決と 5.1 節に示した未実装部の実装を行い、シームレスハンドオーバーを実現する必要があることが分かった。さらにハンドオーバーのトリガを、実験結果として得られた通信断絶時間のばらつきを考慮して決定する必要があることが分かった。

また本提案のハンドオーバートリガは電波強度のみを想定しているが、電波強度のみをハンドオーバーのトリガとすると、例えばネットワークの混雑などによりスループットが低下しても、電波強度さえよければ接続を続けてしまうと

いう課題が残る。この課題を解決する手法として、ハンドオーバーのトリガとして電波強度とパケット到達間隔の両方をトリガとする手法 [13] や電波強度と Goodput (アプリケーションが達成するスループット) の両方をトリガとする手法 [14] などが提案されている。今後は、電波強度だけでなく回線の混雑状況やスループットを考慮しハンドオーバーのトリガとするなど、より利便性の高いシームレスハンドオーバーシステムを検討する必要がある。

参考文献

- [1] C. Perkins, E.: IP Mobility Support for IPv4, Revised, RFC 5944, IETF (2010).
- [2] C. Perkins, E.: Mobility Support in IPv6, RFC 6275, IETF (2011).
- [3] 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).
- [4] 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).
- [5] IEEE 802.21 Standard, "Local and Metropolitan Area Networks - Part 21: Media Independent Handover Services", IEEE Computer Society (2009).
- [6] R. Koodli, E.: Mobile IPv6 Fast Handovers, RFC 5568, IETF (2009).
- [7] 出水達也, 相田 仁: IEEE 802.11n に基づくシームレスなハンドオーバーの実験的実装と評価, 電子情報通信学会技術研究報告. NS, ネットワークシステム, Vol. 112, No. 463, pp. 95-100 (2013).
- [8] 三屋光史郎, 北地三浩, 長澤知津子, 守田空悟, 横田知好, 湧川隆次, 村井 純: IEEE802.21 を用いたスムーズな異種メディア間ハンドオーバーシステムの実現, 情報処理学会論文誌, Vol. 49, No. 1, pp. 335-349 (2008).
- [9] Devarapalli, V., Wakikawa, R., Petrescu, A. and Thubert, P.: Network Mobility (NEMO) Basic Support Protocol, RFC 3963, IETF (2005).
- [10] 福山陽祐, 上醉尾一真, 鈴木秀和, 旭 健作, 内藤克浩, 渡邊 晃: Android 端末における Wi-Fi/3G 間のシームレスハンドオーバーの提案と実装, 情報処理学会研究報告. MBL, [モバイルコンピューティングとユビキタス通信研究会研究報告], Vol. 2013, No. 27, pp. 1-8 (2013).
- [11] Corujo, D., Guimares, C., Santos, B. and Aguiar, R. L.: Using an Open-Source IEEE 802.21 Implementation for Network-Based Localized Mobility Management, *IEEE Communications Magazine*, Vol. 49, No. 9, pp. 114-123 (2011).
- [12] Kamienuo, K., Suzuki, H., Naito, K. and Watanabe, A.: Development of mobile communication framework based on NTMobile, *Mobile Computing and Ubiquitous Networking (ICMU)*, pp. 27-32 (2014).
- [13] 北爪竜馬, 山本 潮, 小野里好邦: パケット到着間隔によるハンドオーバートリガを用いた異種無線システム切替方式, 電子情報通信学会技術研究報告, Vol. 111, No. 261, pp. 49-56 (2011).
- [14] Busanelli, S., Martalo, M., Ferrari, G. and Spigoni, G.: Vertical Handover between WiFi and UMTS Networks: Experimental Performance Analysis, *International Journal of Energy, Information and Communications*, Vol. 2, No. 1, pp. 75-96 (2011).