

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

福山 陽祐¹ 上醉尾 一真¹ 鈴木 秀和¹ 旭 健作¹ 内藤 克浩² 渡邊 晃¹

概要: スマートフォンの普及や無線技術の発展により、移動中にネットワークを切り替えをしたいという要望が高まっている。しかし、この要望を満たすためには、移動透過性の実現と NAT 越え問題を解決しなければならない。筆者らは、これらの課題を同時に解決する NTMobile (Network Traversal with Mobility) を提案し、実装を行っている。NTMobile は Linux ベースの PC 向けに開発が進められており、Android 端末への移植と動作検証を終えている。しかし、ネットワークの切り替え時に通信断絶時間が発生し、通信が再開されるまでに時間がかかるという課題があった。そこで本稿では、Android 端末の 3G と Wi-Fi インターフェースを同時に動作させることにより通信断絶時間をなくし、パケットロスが発生しないハンドオーバー方法について提案する。提案方式を実装し、動作検証した結果、パケットロスのないシームレスハンドオーバーができることを確認した。

キーワード: 移動透過性, シームレスハンドオーバー, Android

Proposal of Seamless Handover between Wi-Fi and 3G in Android Terminals and Its Implementation

FUKUYAMA YOSUKE¹ KAMIENOO KAZUMA¹ SUZUKI HIDEKAZU¹ ASAHI KENSAKU¹
NAITO KATSUHIRO² WATANABE AKIRA¹

Abstract: With the development of wireless technology, there is an increasing desire to switch the network on the move. However, it is necessary to solve the NAT traversal problems and realize mobility at the same time. We have been proposing the technology called NTMobile (Network Traversal with Mobility) that can deal with the subjects. NTMobile is implemented in Linux-based PC and we have finished porting and verified in Android systems. However, there is a problem that there is a certain break time when network switches. In this paper, we propose a way to switch the network seamlessly between 3G and Wi-Fi eliminating communication break time in Android terminals.

Keywords: Mobility, Seamless Handover, Android

1. はじめに

Android OS を搭載したスマートフォンをはじめとする高機能携帯端末の普及に伴い、移動しながら通信を行いたいという要望が高まっている。Android 端末などのスマー

トフォンの多くは、携帯電話網の通信インターフェース (以下 3G) による通信と、無線 LAN (以下 Wi-Fi) による通信の両方が利用可能である。3G はほぼ全国のエリアで使用することができるが、通信の帯域が狭く、データ通信の速度が遅い。一方、Wi-Fi は範囲は狭く限定されるが、高速なデータ通信が可能である。3G の通信中であっても、Wi-Fi が利用できる環境であれば、Wi-Fi に切り替える方が高速に通信ができるため利便性が高い。逆に、Wi-Fi の使用中に移動する場合にも、Wi-Fi の電波が届かなくなっ

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

たら、即座に 3G に切り替えを行った方がよい。

しかし、データ通信に用いられる IP ネットワークでは、端末の通信インターフェースに割り当てられている IP アドレスを用いて通信を管理している。端末の移動に伴い、通信インターフェースを切り替えた場合に IP アドレスが変化する。そのため、アプリケーションは違う通信と認識し、それまで行っていた通信を継続することができない。このような課題を解決する技術を移動透過技術と呼び、現在までに様々な移動透過技術が提案されてきた [1-4]。

また、現在普及している IPv4 ネットワークでは、IPv4 グローバルアドレスの枯渇対策として 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る。しかし、NAT の導入により、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して通信を開始できない NAT 越え問題と呼ぶ課題があり、IPv4 の汎用性を損なう要因になっている。これらの問題は、IPv6 環境への移行により解決が可能であるといわれているが、IPv6 は IPv4 と互換性がなく、IPv6 は今だ普及には至っていない。よって、今後も IPv4 環境が主に使用されていくことが考えられ、NAT 越え問題の解決は重要である。

そこで筆者らは、移動透過と NAT 越えを同時に実現する NTMobile (Network Traversal with Mobility) [5,6] を提案してきた。NTMobile では、NTM 端末 (NTMobile の機能を搭載した端末) に仮想 IP アドレスを与えることにより、端末の移動に伴う実 IP アドレス (実際に物理インターフェースに割り当てられているアドレス) の変化を隠蔽し、アプリケーション間の通信を継続することができる。また、NAT の有無や IPv4 ネットワーク間、IPv6 ネットワーク間 [7]、IPv4 ネットワークと IPv6 ネットワーク間の混在環境 [8] など、あらゆる通信環境に対応している。特定の状況を除いては、エンドツーエンドで通信を行えるため、経路が冗長になりにくいという特徴がある。実際の用途を考えると、携帯端末としては Android 端末などのスマートフォンが一般であり、NTMobile でも Android 端末での利用を想定し、移植と動作検証を終えている [9]。Android 端末の場合、3G と Wi-Fi を切り替えることになる。

Android 端末の切り替え動作を調べたところ、Wi-Fi から 3G にハンドオーバーする際に大きな通信断絶時間が発生することが分かった。この間、パケットロスが発生してしまうため、NTMobile の実用化において大きな課題となる。通信断絶が発生する理由として、迅速な 3G の復帰ができていないからである。Android 端末は、Wi-Fi 接続時には 3G の接続を切断し、通信インターフェースをダウンさせている。そのため、Wi-Fi 切断後 3G の通信準備を行う必要があるため、準備が完了するまでの間通信断絶が発生する。

そこで、本稿では Android 端末の Wi-Fi 接続時に 3G を切断させる動作を見直し、両インターフェースを同時に接

続状態にすることで、切り替え時の通信断絶時間の発生をなくす。3G ネットワークは、どこでも通信できるという特徴に着目し、Wi-Fi 接続時にも 3G は常に通信可能状態にしておき、通信インターフェースを切り替える前に NTMobile を実行し、トンネルを準備しておくことで、ハンドオーバー時のパケットロスを発生させない方式を提案する。その際、アプリケーションの通信と NTMobile の制御メッセージを同時に、同じ相手に異なる通信インターフェースから送信する必要がある。しかし通常、同じ相手に通信インターフェースを分けてパケットを送信することはできない。それは、ルーティングテーブル 1 つに対してデフォルトゲートウェイが 1 つしか設定できないためである。そこで、通信インターフェースごとにルーティングテーブルを用意し、どちらのルーティングテーブルを参照するかを制御することにより、これを可能にした。提案方式を Android 端末に導入して動作確認により、切り替え時にパケットロスのないシームレスなハンドオーバーが可能であることを確認した。

以下、2 章で本提案で使用する移動透過技術である NTMobile とターゲットである Android 端末の通信インターフェース切り替え動作とその課題を述べる。3 章で提案方式を示し、4 章で実装方法について述べる。5 章で動作検証の結果を示し、6 章でまとめる。

2. 要素技術

本章では、移動透過プロトコルとして本研究で利用する NTMobile について説明する。また、実装ターゲットである Android 端末固有のハンドオーバーの課題についても整理する。

2.1 NTMobile

NTMobile で想定するネットワークを図 1 に示す。NTMobile は DC (Direction Coordinator), NTM 端末, RS (Relay Server) によって構成される。

DC は仮想 IP アドレスの割り当て管理や、NTM 端末に対してトンネル構築などの指示を出す装置である。NTM 端末に割り当てられる仮想 IP アドレスは、一意なアドレスになるように DC がアドレスを管理している [10]。NTM 端末は移動先のネットワークから割り当てられる実 IP アドレスと、DC から割り当てられる仮想 IP アドレスの 2 つのアドレスを保持している。NTM 端末が使用するアプリケーションは、仮想 IP アドレスに基づいた通信を行うことにより、NTM 端末の実 IP アドレスが変化しても通信を継続することができる。仮想 IP アドレスに基づくアプリケーションパケットは、NTM 端末間に構築される UDP トンネルによって転送される。図 1 における NTM 端末 A と移動前の NTM 端末 B のように、どちらか一方の端末がグローバルネットワークに接続している場合には、エンド

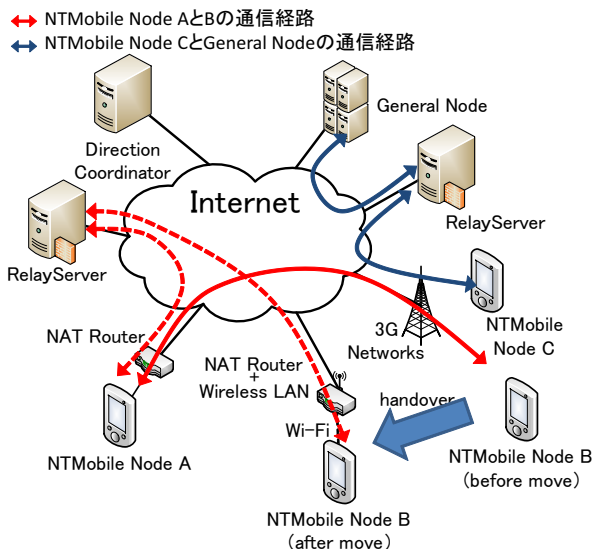


図 1 NTMobile の概要

ツーエンドでトンネルが構築される。RS は、図 1 における NTM 端末 C と一般サーバ (General Node) のように通信相手が NTMobile を実装していない端末と通信する場合や、両 NTM 端末が NAT 配下に位置する場合に、通信の中継を行う装置である。

NTMobile では同様の仕組みを用いて IPv6 や、IPv4/IPv6 の混在環境にも対応することができるが、本稿では IPv4 ネットワークに接続した NTM 端末同士が通信を行う際の動作のみを記述する。

2.1.1 動作シーケンス

NTMobile の動作シーケンスは以下の 3 つのフェーズを経て通信を開始する。

- (1) 端末情報の登録
- (2) 名前解決処理
- (3) トンネル構築

(1) は、NTM 端末起動時および端末移動時に実 IP アドレスなどの端末情報を DC に登録し、仮想 IP アドレスを取得するフェーズである。この際、NAT の有無の情報も登録される。次に (2) において、MN (Mobile Node) は通信相手 CN (Correspondent Node) の実 IP アドレスや仮想 IP アドレスを知るために DC に名前解決を依頼する。MN と CN は NTMobile を実装している NTM 端末である。NTMobile では、このフェーズをトリガーとして図 2 に示す NTMobile 特有のネゴシエーションである (3) を DC から得られた情報をもとに実行する。(3) フェーズ完了後、通信を開始する。

(3) において、DC の指示に従い通信相手との間にトンネルを構築するためのメッセージ交換が行われる。図 2 にトンネル構築までのメッセージ交換の流れを示す。ここでは MN がプライベートネットワークに存在し、CN がグローバルネットワークに存在する場合のトンネル構築を説

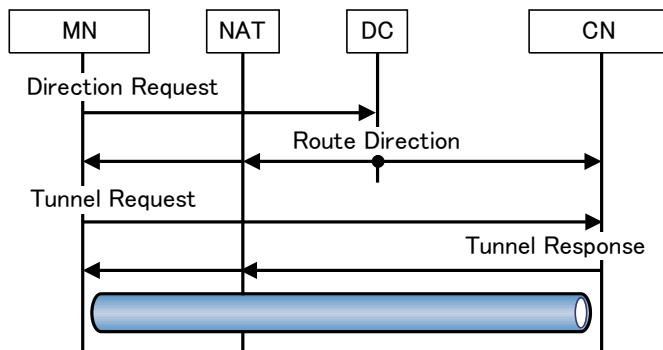


図 2 フェーズ (3) : トンネル構築手順

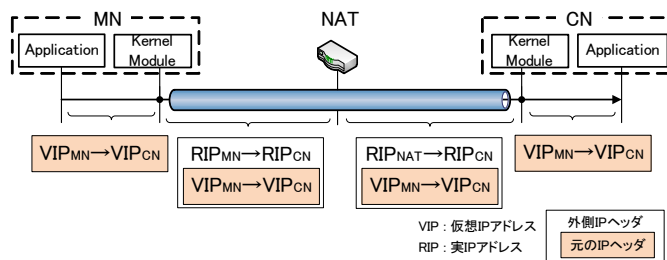


図 3 トンネル通信時のアドレス遷移

明する。

MN は、DC へ Director Request を送信し、トンネル構築の指示を要求する。DC は MN と CN 間で直接トンネルを構築するために、MN へ Route Direction を送信し、CN 宛に Tunnel Request を送信するように指示する。また、DC は CN に対して MN から送信された Tunnel Request を受信するように指示する Route Direction を送信する。Route Direction を受け取った MN、CN は DC の指示に従い、Tunnel Request/Response を交換し、トンネル構築が完了する。Tunnel Request をプライベートネットワーク側の端末から送信することで、NAT に MN と CN が通信を行うためのマッピング情報が生成され、MN と CN の間に NAT をまたがったトンネルを構築することができる。

2.1.2 トンネル通信

MN が CN へトンネル通信を行う様子を図 3 に示す。アプリケーションレベルでは、仮想 IP アドレスに基づいた通信が行われるため、アプリケーションが生成したパケットには仮想 IP アドレス VIP が記載される。MN は宛先アドレスである VIP_{CN} をカーネルモジュールにてフックし、実 IP アドレス RIP_{CN} でカプセル化を行い、CN へ送信する。カプセル化の際には IP ヘッダと UDP ヘッダ、NTMobile 特有の NTM ヘッダが付加される。CN はカプセル化されたパケットを受信すると、カーネルモジュールにてデカプセル化を行い、元の VIP_{CN} 宛のパケットを抽出する。その後、抽出したアプリケーションパケットを上位アプリケーションへ渡す。図 3 のように通信経路上に NAT が存在しても、アドレスが変換されるのは RIP のみ

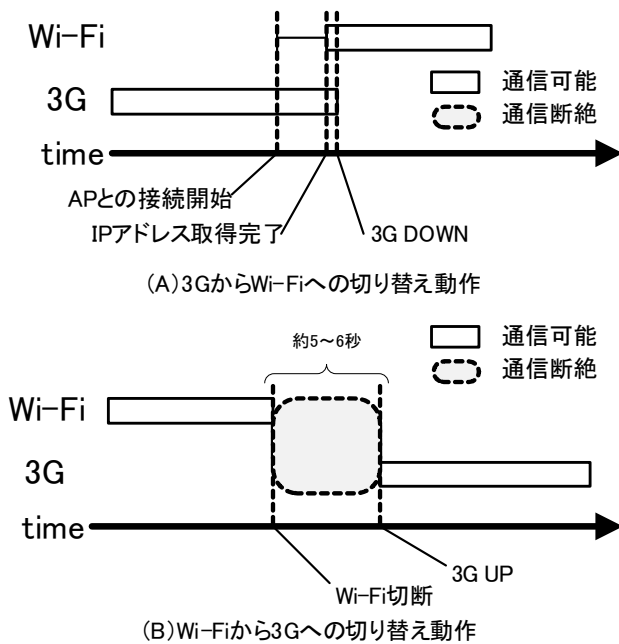


図 4 Android 端末の切り替え動作

であるので、MN と CN のアプリケーションは NAT に影響されることなく通信を行うことができる。

2.1.3 ネットワーク切り替え時の動作

MN の移動や無線インターフェースの切り替えにより実 IP アドレスが変化した際には、CN との間にトンネルを再構築する。この時、MN は CN の実 IP アドレスなどの情報は取得済みであるため、名前解決処理を省略して、トンネル構築処理を実行する。MN のアプリケーションは、仮想 IP アドレス宛にパケットを生成するため、実 IP アドレスの変化の影響を受けることはない。

2.2 Android 端末の通信インターフェース切り替え動作と課題

NTMobile により、移動透過は実現できるが、Android 端末において通信インターフェースを切り替える際、現状の切り替え動作では、通信断絶時間が大きく、そのままでは実用的ではないことが、実際に切り替えの挙動を調査し、判明した。図 4 にそれぞれの切り替え動作とそれに伴う通信断絶時間の関係を示す。

切り替え動作としては、3G から Wi-Fi に切り替え (図 4 の (A)) を行う際には、Wi-Fi 側の IP アドレスの取得が完了すると 3G での接続が解除され、インターフェースがダウンするため、通信断絶はない。だが、NTMobile のトンネル構築処理が開始されるのは、3G インターフェースがダウンした後であるため、トンネル構築が完了するまでの間、パケットロスが発生する。

一方、Wi-Fi から 3G への切り替え時 (図 4 の (B)) には、Wi-Fi 接続が切断されたことを Android OS の通信管理システムが検出すると、3G インターフェース側が接続準備を開始する。この接続処理には多くの時間を要し、Wi-Fi の切断から 3G の接続が完了するまでの 5~6 秒程度の間、通信が全くできなくなる。パケットロスは、この通信断絶時間とトンネル構築にかかる時間を合わせた時間分だけ発生する。

このように Android 端末では、特に Wi-Fi から 3G への切り替え時に長い通信断絶が発生し、多くのパケットロスが発生することが予想される。仮に NTMobile を用いて移動透過を実現できたとしても、上記課題を解決しなければシームレスなハンドオーバーの実現はできない。

3. 提案方式

3.1 提案の方針

本提案方式では、Wi-Fi 接続時にも 3G の接続を切断させないようにして、接続状態を維持することで、2.2 節の課題を解決する。ハンドオーバー手法として、一方の通信インターフェースで通信中に、もう一方のインターフェースで NTMobile のトンネルを準備する手法を採用する。この手法により、トンネルを切り替えるのみでインターフェースの切り替えが実現でき、理論的にはパケットロスは発生しない。

しかし通常、同じ通信相手に対して、同時に複数のインターフェースから経路を分けてパケットを送信することはできない。ルーティングテーブル 1 つに対して、デフォルトゲートウェイが 1 つしか設定できず、送信インターフェースが固定されてしまうためである。提案するハンドオーバー手法を行うためには、同じ相手にアプリケーションのトンネル通信と NTMobile のトンネル構築処理を別々のインターフェースを用いて同時に実行する必要がある。そこで、iproute2^{*1}の仕組みを用いてパケットのルーティングを行うことで、上記の問題を解決し、トンネル通信とトンネル構築処理を同時に行うことを可能にする。iproute2 は、Linux のルーティングテーブルを操作するパッケージで、ルーティングテーブルを複数生成することができ、どのテーブルを参照させるかのポリシーを設定することが可能である。

よって、Wi-Fi インターフェースを送信インターフェースとする Wi-Fi 用のルーティングテーブルと、3G インターフェースを送信インターフェースとする 3G 用のルーティングテーブルを生成し、両インターフェースのルートを確保する。パケットごとに、Wi-Fi と 3G のどちらのルーティングテーブルを使用するかをポリシーとして設定することにより、同時通信を可能にする。

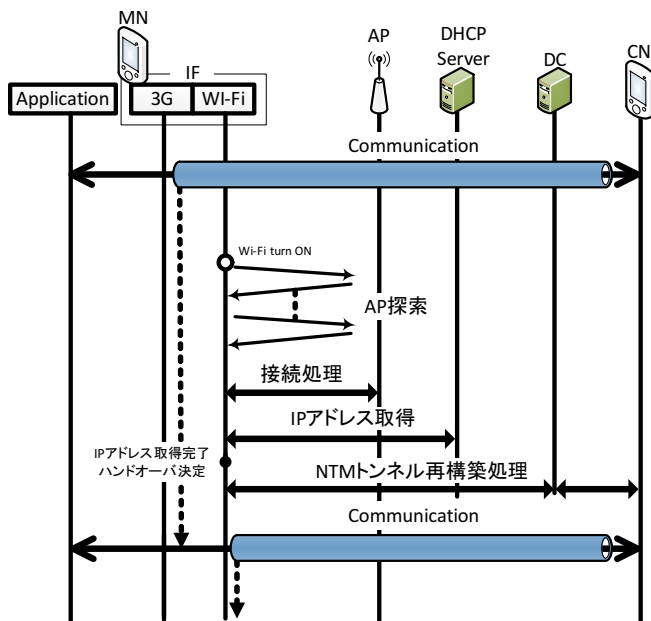


図 5 3G から Wi-Fi へのハンドオーバーシーケンス

3.2 3G から Wi-Fi へのハンドオーバー動作

図 5 に 3G から通信を開始して、Wi-Fi へハンドオーバーする時の動作を示す。MN は、3G 側ですでに NTMobile のトンネル通信を開始しているものとする。MN は、3G 側でのトンネル通信中に、Wi-Fi インターフェース側で無線アクセスポイント（以下 AP）を探索するために、定期的にチャネルスキャンを行い、周囲の AP を検索する。そして、内部に持つ接続可能 AP リストとチャネルスキャンにより得られた周辺 AP を比較し、接続する AP を決定し、コネクションを確立する。接続可能 AP リストとは、WEP (Wired Equivalent Privacy) [11] や WPA (Wi-Fi Protected Access) [12] などで暗号化がされている AP に接続するために事前に AP の名前とパスワードを登録したリストである。もし、接続可能 AP が複数見つかった場合は、電波強度の強い AP を選択する。また、接続可能な AP が見つからない場合には Wi-Fi を OFF にし、一定時間後に再度 Wi-Fi を ON にして周辺の AP を探索しなおす。

MN は AP に接続後、DHCP のシーケンスにより IP アドレスを取得し、Wi-Fi に IP アドレスが割り当てられることにより、Wi-Fi の通信準備を完了する。その後、Wi-Fi 側からトンネル再構築処理を行い、3G 側のトンネルから Wi-Fi 側のトンネルにパケットの流れを切り替えることにより、ハンドオーバーを完了する。

3.3 Wi-Fi から 3G へのハンドオーバー動作

図 6 に Wi-Fi から通信を開始して、3G へハンドオーバーする時の動作を示す。MN は、Wi-Fi 側で NTMobile のト

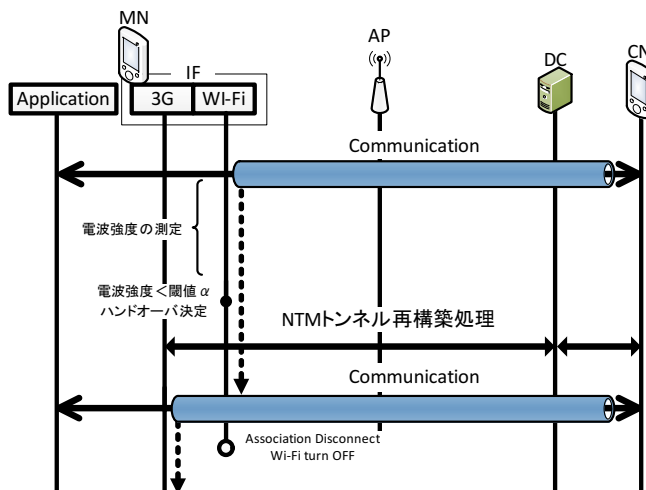


図 6 Wi-Fi から 3G へのハンドオーバーシーケンス

表 1 使用するポリシー

	対象パケット	参照先ルーティングテーブル
(A)	カプセル化パケット	Table 3G
	その他のパケット	Table Wi-Fi
(B)	全てのパケット	Table Wi-Fi
(C)	カプセル化パケット	Table Wi-Fi
	その他のパケット	Table 3G
(D)	全てのパケット	Table 3G

ンネル通信をすでに行っているものとする。Wi-Fi は、通信中に AP との電波強度を測定し、通信品質の監視を行う。AP の電波強度は、AP から送信されるビーコンや、データパケットを受信した際に取得することができる。3G は常時接続状態にしておき、AP の電波強度が閾値 α より下回った場合、通信状態が不安定になったと判断し、3G 側へのハンドオーバーを決定する。

3G 側へのハンドオーバーを決定すると、3G 側からトンネル再構築を行い、3G 側にトンネルを生成する。Wi-Fi 側のトンネルから 3G 側のトンネルにパケットの流れを切り替えることで 3G 側へのハンドオーバーを完了する。ハンドオーバー完了後、Wi-Fi のコネクションを切断し、Wi-Fi を OFF にする。

3.4 iproute2 を用いたパケットルーティング

表 1 に設定するポリシーを表記し、ポリシーの設定タイミングを図 7 に示す。図 7 は、ハンドオーバーを決定してからトンネルを切り替えるまでの動作を示し、図中の点線より上側が 3G から Wi-Fi にハンドオーバーする際の動作で、下側が Wi-Fi から 3G にハンドオーバーする際の動作である。

3G から Wi-Fi へのハンドオーバー時には、ハンドオーバーを決定すると Wi-Fi 用のルーティングテーブルを生成する。3G 用のルーティングテーブルは、通信開始前に生成しておく。この時点で、3G 用のルーティングテーブルと Wi-Fi 用のルーティングテーブルの 2 種類が存在すること

*1 Linux Advanced Routing & Traffic Control HOWTO,
<http://linuxjfn.sourceforge.jp/JFdocs/Adv-Routing-HOWTO/index.html>

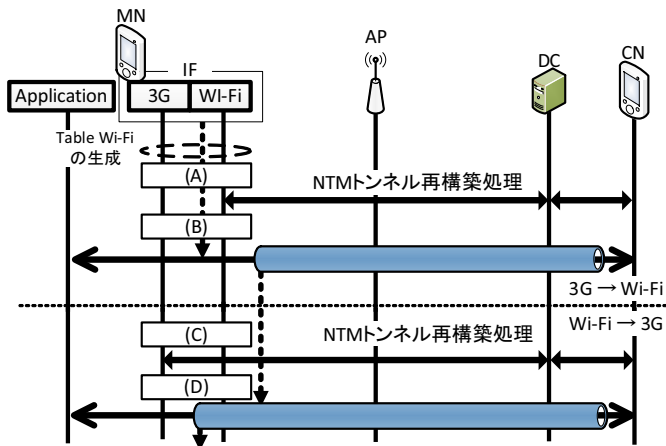


図 7 ポリシー設定タイミング

になる。これらのテーブルを Table 3G と Table Wi-Fi と表記する。Table Wi-Fi を生成の後、(A) のポリシーを設定する。(A) では、今までの通信を続けるために、カプセル化パケットを Table 3G を、NTMobile のトンネル再構築を Wi-Fi で行うため、カプセル化パケット以外を Table Wi-Fi を参照させる。そして、NTMobile のトンネル再構築処理を行い、完了後 (B) のポリシーを設定し、全てのパケットを Wi-Fi から送信するようにする。(B) のポリシーにより、3G 側から Wi-Fi 側にトンネルの切り替えを行う。

Wi-Fi から 3G へのハンドオーバー時には、通信開始時は (B) の設定によりすべて Wi-Fi から送信されている。電波強度の低下により、3G へのハンドオーバーを決定した際は、(C) のポリシーを設定しカプセル化パケットは Table Wi-Fi を参照し通信を Wi-Fi で続け、NTMobile のトンネル再構築を 3G で行うため、カプセル化パケット以外を Table 3G を参照するようにする。その設定後、NTMobile のトンネル再構築処理を行い、完了後 (D) のポリシーを設定し、全てのパケットを 3G から送信するようにする。(D) のポリシーにより、Wi-Fi 側から 3G 側にトンネルの切り替えを行う。トンネルの切り替え後、Wi-Fi の接続を切断すると、Table Wi-Fi が消滅する。

4. 実装方法

4.1 システム構成

本研究は、Android 端末上で動作させることを目的としている。よって、Android のシステムアーキテクチャが重要になってくる。まずは、Android のシステムアーキテクチャを紹介し、次に提案システムをアーキテクチャのどこに適応させるかを説明する。図 8 に Android のシステムアーキテクチャと提案システムの実装箇所の構造を示す。

Android のシステムアーキテクチャは、以下のコンポーネントにより構成されている。

- カーネル層

Linux Kernel 2.6 以降をベースに実装されている。

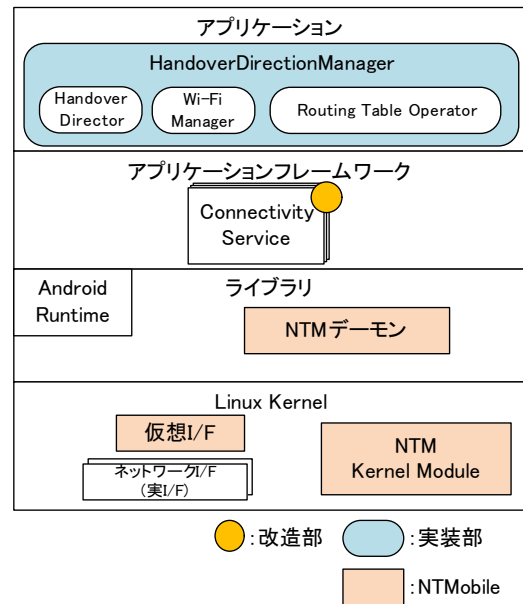


図 8 提案システムのソフトウェア構造

- ライブラリ層

暗号化から描画制御まで、さまざまな機能を提供するライブラリ群が組み込まれている。C/C++ のライブラリも組み込まれており、ネイティブアプリケーションとして動作させることが可能である。

- Android Runtime

Dalvik VM と呼ばれる Java 仮想マシンと、Java ライブラリが実装されている。Android アプリはすべてこの Dalvik VM 上で実行される。

- アプリケーションフレームワーク層

アプリケーションフレームワーク層では主に、アプリケーションの起動から終了までの流れの管理を実施する。さらに、UI の表示やユーザによる操作など携帯端末で発生する状態変化を、各アプリケーションに伝える手段を提供している。アプリケーションで使われる様々な機能を提供する API である。

- アプリケーション層

Google Play*2 で提供されるアプリケーションや電話機能、HOME 画面機能などは全てこの層に位置する。

本提案の前提である、Wi-Fi と 3G の同時接続を実現するために、アプリケーションフレームワーク層内の ConnectivityService に変更を加えた。通常の Android は 2.2 節で示したように、Wi-Fi と 3G が同時に動作することはない。そのため、OS 内のネットワークの接続状況を監視している ConnectivityService を改造することにより、Wi-Fi と 3G を同時に接続状態にすることができる。そのために、Android のソースコード*3 のダウンロードを行い、ConnectivityService に変更を加え、Wi-Fi と 3G が同時に

*2 Google Play, <https://play.google.com/store>

*3 Android Opensource Project, <http://source.android.com/>

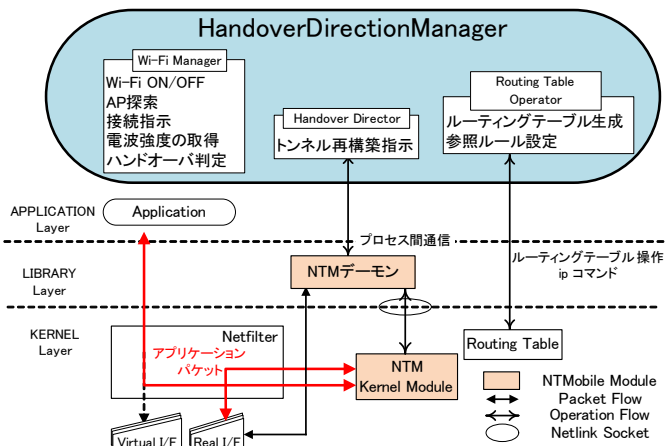


図 9 HDM と NTMobile の関係

接続状態になるカスタム Android OS を作成した。本提案は、このカスタム Android OS に NTMobile と提案ハンドオーバー方式を行う機能を実装する。

実装コンポーネントは、NTMobile モジュールと HandoverDirectionManager (以後 HDM) である。NTMobile は、カプセル化などを行うカーネルモジュールとネゴシエーションを行うデーモンプログラム (以後 NTM デーモン)、仮想インターフェースにより動作する。カーネルモジュールはカーネル層に実装され、NTM デーモンはライブラリ層にネイティブアプリケーションとして実装されている。

提案方式の切り替え手法を行う HDM は、Android アプリとして実装をする。HDM は、Wi-Fi 関連の処理を行う機能 (Wi-Fi Manager) とルーティングテーブルを操作する機能 (Routing Table Operator)、NTM デーモンに指示を出す機能 (Handover Director) の 3 つの機能を持つ。

NTMobile と HDM の関係を図 9 に示す。NTM デーモンは、アプリケーションの名前解決処理を検知すると、名前解決およびトンネル構築を実行する。トンネル構築時のネゴシエーションメッセージは、NTM デーモンより実インターフェースに送信される。一方、アプリケーションが送信するパケットは、Netfilter によりフックされ、カーネルモジュールでカプセル化が行われ、実インターフェースから送信される。カプセル化されたパケットを受信した際にはデカプセル化を行い、アプリケーションへ渡される。HDM は、Routing Table Operator にて iproute2 を実行し、カーネル層のルーティングテーブル情報の操作やパケット参照ポリシーの設定を行う。またハンドオーバーを決定した時に Handover Director から NTM デーモンに対しトンネル再構築処理の開始を指示する。

4.2 HDM の処理内容

HDM は、主に Wi-Fi Manager によって Wi-Fi の制御を行い、ハンドオーバーの決定を行う。まず、Wi-Fi で通信を

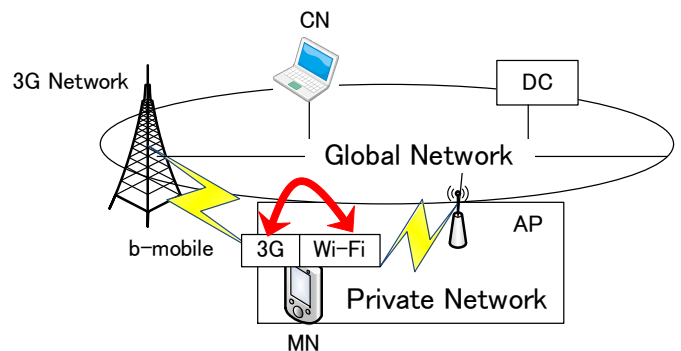


図 10 実験環境

行っているかを判定する。Wi-Fi で通信を行っていない場合は、Wi-Fi デバイスの ON を行い周辺の AP を探すためにチャンネルスキャンを行う。探索により得られた周辺の AP リストとアプリ内に持つ接続可能 AP リストとを比較し、接続可能リストに記載されている AP を見つけた場合、その AP との接続処理要求を Android OS に指示する。もし接続可能 AP が無いもしくは周辺に AP がなければ Wi-Fi を OFF にし、次の実行を待つ。接続可能リストは、Android が保持している接続履歴を取得することにより作成する。

DHCP の処理が完了し、IP アドレスの取得が完了すると、Wi-Fi へのハンドオーバーを決定する。ハンドオーバーが決定されると、Routing Table Operator によりルーティングテーブルの更新とポリシーの設定を行う。さらに、Handover Director により、NTM デーモンに対して Wi-Fi 側でトンネルを再構築する命令を出す。

Wi-Fi 接続が既にされている場合には、AP の電波強度を取得し閾値との比較を行う。閾値を下回った場合、3G へのハンドオーバーを決定し、ルーティングテーブルの更新とポリシーの設定を行う。さらに、Handover Director により、NTM デーモンに対して 3G 側でトンネルを再構築する命令を出す。

Routing Table Operator は、3.4 節で示したルーティングテーブルの操作やポリシーを設定するために、アプリ内から iproute2 を実行する。Handover Director は、NTM デーモンと HDM 間でプロセス間通信を行うことにより、メッセージの送受信を可能にした。

5. 動作検証

Android 端末に ConnectivityService に変更を加えて作成したカスタム OS を実装し、NTMobile と HDM の実装を行った。HDM はテスト用に、アプリ内のボタン操作により任意のタイミングでハンドオーバーが行える機能を実装し、提案方式の動作確認を行った。

5.1 実験環境

図 10 に示す環境でハンドオーバー実験を行った。CN に

表 2 パケットロス数

	NTMobile のみ適応	提案方式
3G ⇒ Wi-Fi	6	0
Wi-Fi ⇒ 3G	55	0

は NTMobile を実装し、グローバル IP アドレスを割り当てている。NTMobile のネゴシエーションに必要な DC もグローバル IP アドレスを割り当てた。MN に、提案方式と NTMobile を実装した。3G ネットワークへの接続は b-mobile SIM のデータ通信専用のものを利用した。3G インターフェースは、グローバル IP アドレスを取得しており、Wi-Fi は AP に接続すると、プライベートアドレスを取得する。

5.2 パケットロスの測定

提案方式の性能を測定するために、5.1 節の環境でハンドオーバーの実験を行った。測定には iperf^{*4}を用い、MN と CN 間で 1470 バイトの UDP パケットを帯域制限 128kbps で送信した。1 回の測定で 20 秒間 UDP パケットを送信し、その間に切り替えの指示を 1 回出す。この測定を 3G から Wi-Fi、Wi-Fi から 3G ともに各 5 回試行した。また、同じ環境において、NTMobile のみを実装した端末でも測定を行い、パケットロスの比較を行った。

表 2 に測定したパケットロス数を示す。NTMobile のみを実装し、ハンドオーバーを行った時のパケットロス数と比較すると、3G から Wi-Fi のハンドオーバーでは、平均 6 パケットのロスが発生しているが、提案方式ではパケットのロスは発生しなかった。Wi-Fi から 3G へのハンドオーバーでも、平均 55 パケットのロスが発生していたが、提案方式ではパケットのロスの発生はなかった。よって、シームレスにハンドオーバーできることを確認した。

5.3 スループットの測定

HDM は、Wi-Fi 接続時に電波強度の測定を頻繁に行うため、Wi-Fi 接続時の電波強度測定間隔が通信に影響を与える可能性が考えられる。そこで、電波強度測定間隔を変化させて、スループットを比較した。HMD の電波強度測定間隔は 10ms, 100ms, 1000ms として実験を行った。測定には、iperf を用い、MN と CN 間で 30 秒間の TCP 通信を 5 回試行してスループットの平均をとった。表 3 にスループット測定結果を示す。どの電波強度測定間隔であっても、スループットにほとんど変化がないことがわかる。このことから、電波強度測定間隔が通信に影響を与えることはほとんどないことを確認した。

6. まとめ

本稿では、Android OS に変更を加えることにより、Wi-Fi

表 3 スループット

測定間隔 [ms]	スループット [Kbps]
10	503
100	496
1000	500

と 3G の同時接続を可能にし、通信断絶時間の発生を除去した。ハンドオーバー手法では、トンネルを事前に準備することで、パケットロスが発生しないハンドオーバー手法を提案した。iproute2 により経路を制御することで、同じ相手に対して一方のインターフェースで通信をしながらもう一方のインターフェースで NTMobile のネゴシエーションを同時に送信することを可能にし、事前にトンネルを構築しておくことを可能にした。動作確認の結果、パケットロスのないシームレスなハンドオーバーが可能であることを確認した。また、HMD が通信に影響を与えることがほとんどないことを確認した。

参考文献

- [1] Le D., Fu X. and Hogrefe D.: A Review of Mobility Support Paradigms for the Internet, IEEE Communications Surveys, Vol. 8, No. 1, pp. 38-51 (2006)
- [2] C. PERKINS: IP mobility support for IPv4, RFC3344, 2002
- [3] D. Johnson, C. Perkins, J. Arkko: Mobility Support in IPv6, RFC3775, 2004
- [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, pp.3244-3257(2006)
- [5] 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, pp.367-379(2013)
- [6] 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 における端末アドレスの移動管理と実装, 情報処理学会論文誌, Vol.54, pp.380-393(2013)
- [7] 上醉尾一真, 鈴木秀和, 内藤克浩, 渡邊 晃: IPv6 ネットワークにおける NTMobile の検討, 情報処理学会研究報告, Vol. 2011-MBL-59, No.9, pp. 1-8 (2011)
- [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, マルチメディア, 分散, 協調とモバイル (DICOMO2012) シンポジウム論文集, Vol.2012, No.1, pp.1169-1179(2012)
- [9] 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.2012-MBL-62, No.19, pp.1-8(2012)
- [10] 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, マルチメディア, 分散, 協調とモバイル (DICOMO2011) シンポジウム論文集, Vol.2011, No.1, pp.1139-1145(2011)
- [11] L. M. S. C. of the IEEE Computer Society: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE Std 802.11, 1999
- [12] IEEE: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Security Enhancements, IEEE Std 802.11i, 2004

*4 入手先 <http://sourceforge.net/projects/iperf/>