平成29年度　卒 業 論 文

和文題目

VPNServiceを利用した移動透過性の実現方式の提案

英文題目

Proposal of Realization Method of IP Mobility using VPNService

情報工学科　渡邊研究室

(学籍番号: 140441050)

黒宮　魁人

提出日: 平成30年2月9日

名城大学理工学部

概要

NTMobile（Network Traversal with Mobility）は，バージョンの異なる IP アドレス間の通信や NAT 越え問題を解決したうえで，移動しながらの通信を可能とする移動透過性を実現することが可能な技術である．NTMobile は当初 LinuxPC を想定して開発された技術であるが，現在は Android 端末に NTMobile を使用するためのアプリケーションが提供され，一般アプリケーションに NTMobile によるバージョンの異なる IP アドレス間の通信や NAT 越えの機能を確認することができた．しかし，移動に関わる部分が実装されておらず，一般アプリケーションに移動透過性を実現できていない．また，NTMobile は定期的にパケットを短い期間で送信する必要があるため，バッテリー駆動のスマートデバイスにそのままの形で組み込むことが難しいという課題がある．そこで本研究では，スマートデバイス向けに提供された NTMobile の移動に関わる処理の提案と実装を行い，NTMobile の定期的なパケットを省略する方式についての提案を行うと同時にスマートデバイスに適した形のシグナリング処理についての再検討を行った．

Abstract

NTMobile (Network Traversal with Mobility) is a technology that can solve the communication between different IP addresses and the NAT traversal problem and realize IP Mobility. NTMobile is a technology originally developed assuming LinuxPC. However, it is now available as an application for using NTMobile for Android terminals. As a result, I was able to confirm the function of communication between NAT traversal and different IP addresses provided by NTMobile for general application. However, since handover processing is not implemented, IP Mobility has not yet been realized for general applications. Also, NTMobile needs to periodically transmit packets in a short period of time. As a result, there is a problem that it is difficult to mount as it is on a battery-operated smart device. In this research, I propose and implement NTMobile handover related processing provided for smart devices, and After proposing to eliminate Keep Alive of NTMobile. In addition, I reexamined the signaling processing suitable for the smart device.

目 次

第1章　序論

スマートデバイスの普及により，モバイルデータトラフィックが爆発的に増加している．最近の調査では 2015 年には 3.7EB/月であったモバイルデータトラフィックが 2020 年までに 30.6EB/月まで増加すると予測されている [1]．そのため，モバイルデータ通信網から IP ネットワークにデータオフロードすることが求めらている．しかし，現在のネットワークは以下のような課題がある．IP ネットワークは IPv4 アドレスと IPv6 アドレスが混在しており，バージョンの異なるアドレス同士で直接通信することができない．また，IPv4 ネットワークではインターネット側から NAT（Network Address Translation）配下の端末に通信を開始することが出来ないといった問題も存在する（NAT 越え問題）．さらに，IP ネットワークでは IP アドレスが位置識別子と通信識別子の 2 つの役割を担っているため，端末の移動に伴う IP アドレスの変化によってそれまで端末が行なっていた通信が切断されてしまう．そのため，移動の激しいスマートデバイスにストレスなくデータオフロードをさせるためにはこの課題の解決が必須である．

これらの課題を解決する技術として筆者らは NTMobile（Network Traversal with Mobility）を提案している．[2] [3] [4] [5]NTMobile とは，NAT 越え問題を解決しつつ移動透過性を実現する通信技術である．まず，NTMobile では NTMobile を導入した端末（以降，NTM 端末）に，位置に依存しない仮想 IP アドレスを割り当てる．NTM 端末に実装されたアプリケーションは割り当てられた仮想 IP アドレスを利用して通信を行うが，実際の通信は端末が割り当てられている実 IP アドレスを用いてカプセル化 / デカプセル化を行う．このため，実 IP アドレスが変化しても通信を継続することが可能である．また，NTM 端末とグローバル空間に設置された NTMobile による通信経路指示を行う装置が定期的に UDP による通信（以降，KeepAlive）を行うことで，端末が NAT 配下に存在していても通信経路指示が可能となり，NAT 越え問題を解決することが出来る．更に，IPv4 と IPv6 間の通信や，通信を行う両端末が NAT 配下に存在するような直接通信ができない環境であっても，中継装置を利用したカプセル化通信を行うことで，通信を実現する．このように現在のインターネットの課題を解決するにあたり NTMobile は非常に有用性の高い技術であるが，スマートデバイスに実装するためには大きな課題が 2 つ存在する．

まず，NTMobile は NAT 越え問題を解決するために 10 秒に 1 度 KeepAlive の送信を行っている．このような短い間隔で行う KeepAlive は PC では問題ないが，バッテリ駆動のスマートデバイスにそのままの形で実装することは難しい．次に，NTMobile ではカーネル空間にてパケットのカプセル化 / デカプセル化を実行するため，NTMobile による通信を行うために端末の管理者権限を取得する必要がある．このため端末の root 化が推奨されていないスマートデバイスに普及させることが難しい．この課題を解決するため，VPN 技術が提供するパケットのカプセル化 / デカプセル化を利用し，NTMobile のためのカプセル化 / デカプセル化として VPN 技術を利用する方法

（VPNService 型 NTMobile）が提案されている．[6] 現在，VPNService 型 NTMobile を使用して一般アプリケーションに NAT 越えや異なるバージョンの IP アドレスによる相互通信などの一部の機能を確認することが出来た．しかし，NTMobile 通信を行うライブラリが OS によって異なるインタフェースを共通して検出することが出来ないため，アドレス変化検出機能が実現できていない．そこで本研究では VPNService 型 NTMobile のアドレス変化検出を NTMobile 通信を行うライブラリから独立させることで移動透過性の実現を行う．また，VPNService 型 NTMobile の省電力化を行うためにプッシュ通知機能である FCM（Firebase Cloud Messaging）[7] を利用し NTMobile が行う KeepAlive を省略する方式について提案する．さらに，FCM を利用したシグナリング処理を行う際に，ネットワーク構成によってはシグナリング処理を簡単化することが可能となるため，シグナリング処理の再検討を行った．以降，2 章で NTMobile について述べ，3 章でプッシュ通知機能について説明する．4 章では提案方式について説明し，5 章で実装について述べた後，最後に 6 章でまとめる．

第2章　NTMobile

本章では，移動透過性を実現したうえで，NAT 越え問題の解決や異なるバージョンの IP アドレスを使用した通信を実現する技術である NTMobile について説明する．

2.1　NTMobile 概要

図 1 に NTMobile の構成を示す．NTMobile は NTM 端末の他に，端末情報の管理や通信経路の指示，仮想 IP アドレスの割り当てを行う DC（Direction Coordinator）及び IPv4/IPv6 間の通信や，NTM 端末が異なる NAT 配下に存在する際に通信の中継を行う RS（Relay Server）によって構成される．DC，RS は Dual Stack Network に配置されていることが前提である．また，ネットワークの規模に応じて，複数台設置することにより負荷分散が可能である．NTM 端末は起動時に DC に登録処理を行うことにより，DC から仮想 IP アドレスの割り当てを受ける．NTM 端末のアプリケーションは，割り当てられた仮想 IP アドレスを用いて通信を行う．NTMobile では，IP アドレスの位置識別子の役割を実 IP アドレスが持ち，通信識別子の役割を仮想 IP アドレスが持つ．通信を行う際は仮想 IP アドレスを利用して作成したパケットを実 IP アドレスにて UDP でカプセル化する．これにより NTM 端末の実 IP アドレスが変化しても通信識別子としての役割を持つ仮想 IP アドレスは変化しないため，移動透過性を実現できる．また，NTM 端末と DC は一定時間ごとに行われる Keep Alive によって，NTM 端末が NAT 配下に存在しても DC からの指示はいつでも受信できる．NTM 端末同士は原則直接通信をするように設計されているが，IPv4/IPv6 間の相互通信の場合，もしくは NTM 端末が異なる NAT 配下に存在する場合は RS が通信の中継を行う．ただし，後者の場合，NAT の種類によっては通信を RS で中継せずに直接通信に切り替えることができる [8].

2.2　NTMobile の動作シーケンス

以降の説明では，通信開始側の NTM 端末を MN（Mobile Node），通信相手側の NTM 端末を CN（Correspondent Node）とする．図 2 に NTMobile の通信開始時のシーケンスを示す．図 2 では，MN，CN は異なる NAT 配下に存在している．また，簡略化のため DC，RS は 1 台としている．前提として，DC には MN と CN の端末情報が既に登録されているものとする．また，定期的な Keep Alive が行われている．

通信開始時，MN は DC に経路指示要求として Direction Request を送信する．DC は受信した Direction Request と，登録済みの CN の情報を確認し，MN と CN が共に NAT 配下であることか

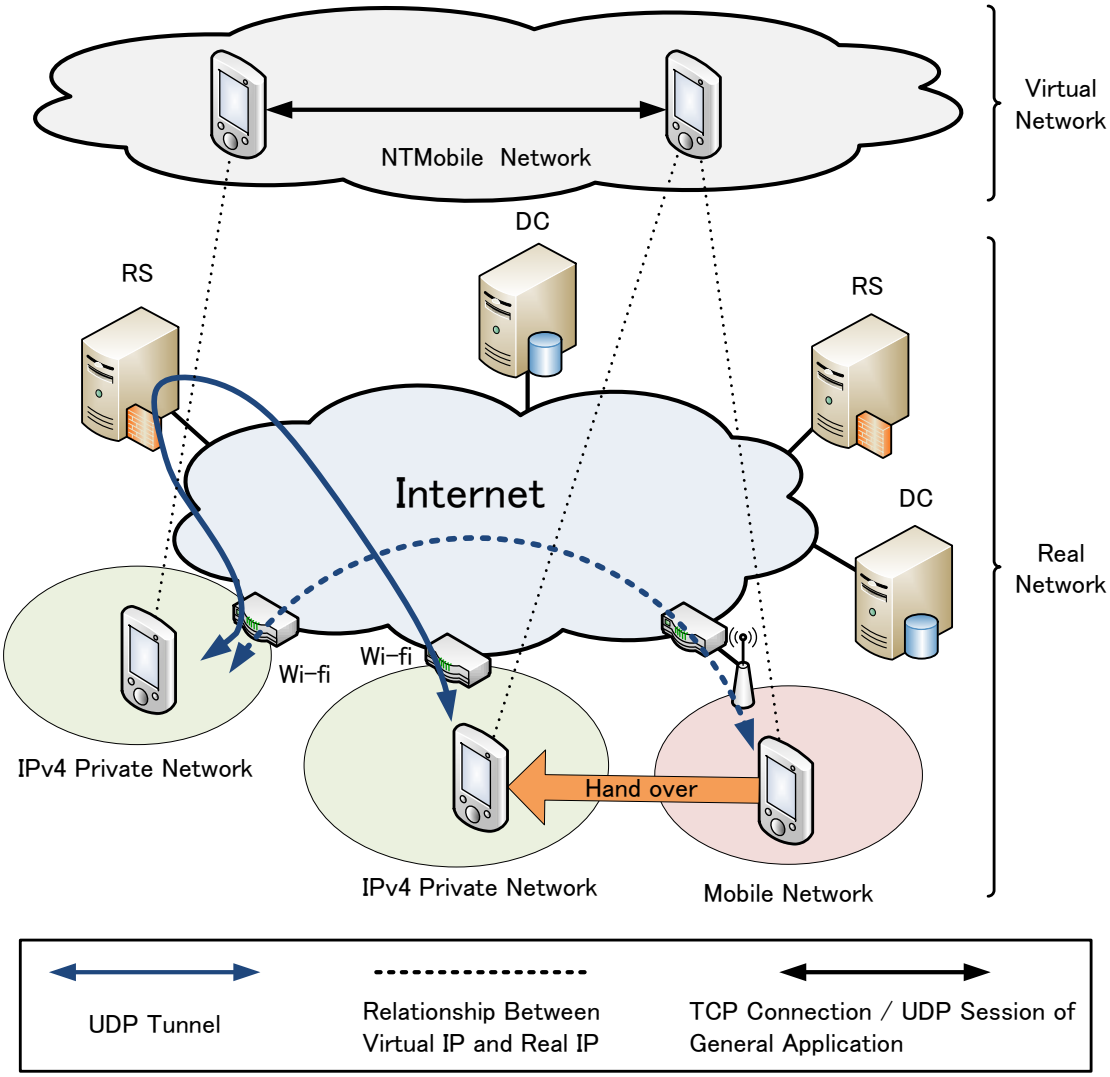

図 1　NTMobile の構成

ら RS を経由した通信経路を構築する必要があると判断する．DC は RS に通信の中継要求である Relay Direction を送信し，RS は ACK を返信する．次に DC は MN と CN に経路指示として Route Direction を送信する． Route Direction を受信した CN は NTMobile 通信に使用するポートを開放するため RS に Hole Punch を送信する．MN は UDP によるトンネルを構築するために Tunnel Request を RS を経由し CN に送信する．Tunnel Request を受信した CN は Tunnel Response を RS を経由して MN に返信する．これにより MN と RS 間の UDP トンネルと RS と CN 間の UDP トンネルが構築される．

2.3　NTMobile の自律的経路最適化

図 3 に NTMobile における自律的経路最適化のシーケンスを示す．[8] 図 3 の NAT_{MN} と NAT_{CN} はともに Port Restricted NAT である時のシグナリング処理である．また，2.2 節にて説明した Direction

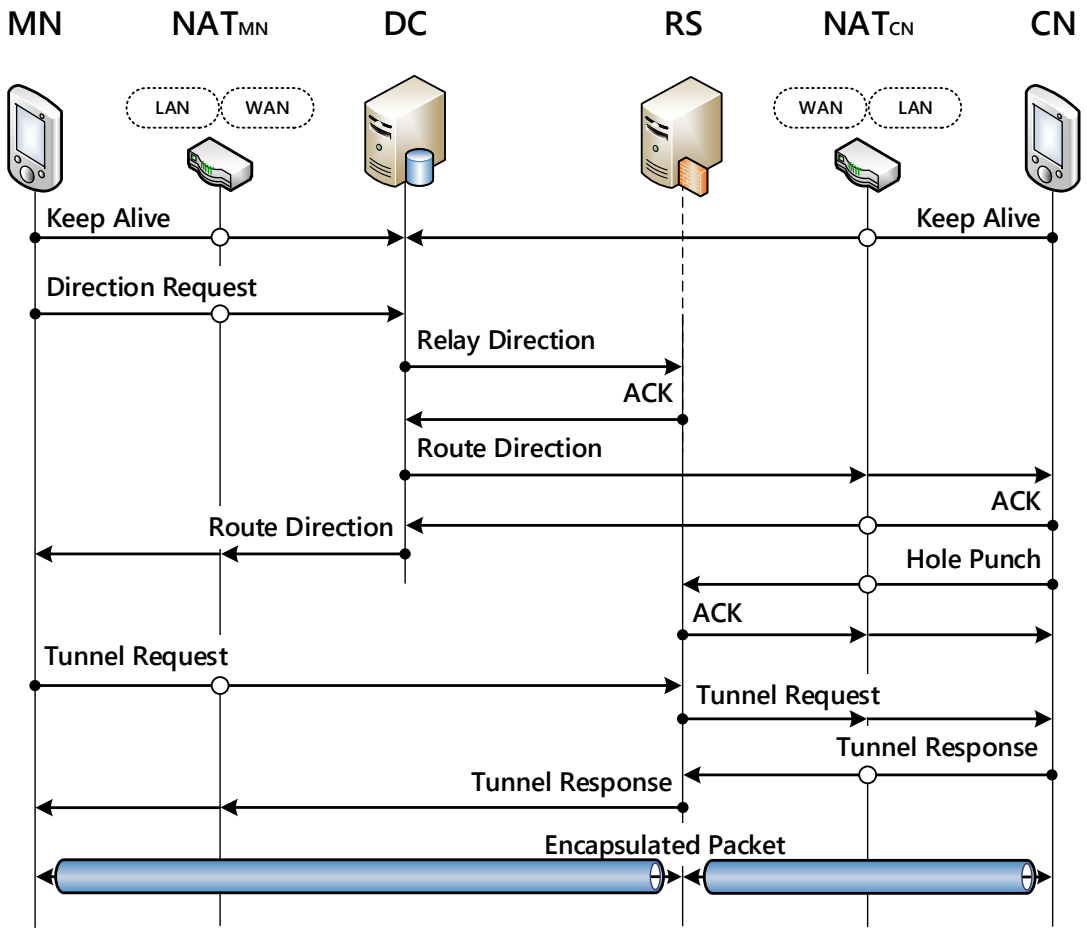

図 2　NTMobile の通信開始時のシーケンス

Request から Tunnel Response までのシーケンスは NTMobile Signaling として記述している．

NTMobile の自律的経路最適化では，まず DC の指示通りに RS 経由の通信経路を構築する．次に，RS 経由の通信を行いながら，MN と CN が Autonomous Tunnel Request を通信相手の NAT に向けて複数回投げ合う．もし，一方の Autonomous Tunnel Request が通信相手の NTM 端末に到達すれば最適経路が存在すると判断できる．Autonomous Tunnel Request を受信した NTM 端末は通信相手に Autonomous Tunnel Response を返信する．Autonomous Tunnel Response の到達が確認できた場合，トンネル経路を RS を経由しない直接通信の経路に切り替える．いずれの NTM 端末も Autonomous Tunnel Response を受信できなかった場合は，RS を経由しないと通信ができないと判断し，既に構築されている RS を経由したトンネル通信を継続する．自律的経路最適化を行うために MN は NAT_{CN} の IP アドレスとポート番号，CN は NAT_{MN} の IP アドレスとポート番号の情報が必要である．これらの情報を NTM 端末に伝えるために，DC は MN 宛の Route Direction に NAT_{CN} のポート番号を，CN 宛の Route Direction に NAT_{MN} のポート番号を追記して送信する．Route Direction に記述されている情報に従い MN と CN は Autonomous Tunnel Request を通信相手の NAT に送信し，自律的経路最適化を試みる．このとき Port Restricted NAT のような制約があ

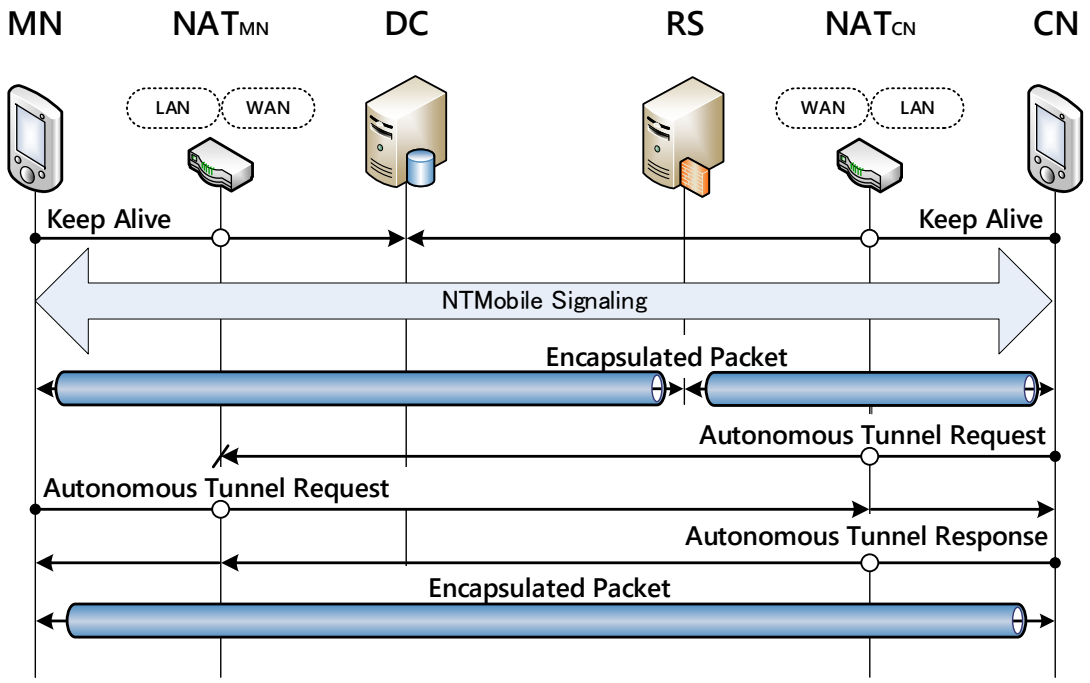

図 3　NTMobile の自律的経路最適化のシーケンス

る程度強い NAT では，通信相手との NAT エントリが作成されるまでパケットが廃棄されるため複数回 Autonomous Tunnel Request を投げ合う必要がある．文献 [8] で提案されている手法では，Autonomous Tunnel Request を MN，CN が 3 回ずつ投げ合うことで，パケットの到達性があるかどうかを確認している．また，片側の NAT が最も制約の弱い Full Cone NAT である場合は，最初に送信した Autonomous Tunnel Request が廃棄されることなく相手に到達する．

2.4　NTMobile の移動通信シーケンス

図 4 に移動に係る NTMobile の通信シーケンスを示す．図 4 では，カプセル通信を行っている途中で CN がアドレス変化した場合を想定したシーケンスである．NTMobile では，アドレスの変化が検出できた場合は，再度 DC に実 IP アドレスの登録作業（Registration）を実行した後に再度シグナリングを行うことで，通信の継続が可能となる．

2.5　NTMobile Framework（NTMfw）

NTMobile 通信を提供する C で記述された通信ライブラリである NTMobile Framework（NTMfw）がある．NTMfw をアプリケーション内に組み込むことにより，NTMobile の機能を root 権限無く利用することが可能となる．また，パケットのカプセル化 / デカプセル化には lwip（A Lightweight TCP / IP stack）というユーザ空間における仮想 IP スタックを利用することで実現している．NTMfw により一般のスマートデバイスに NTMobile による通信を提供することが可能となったが，アプ

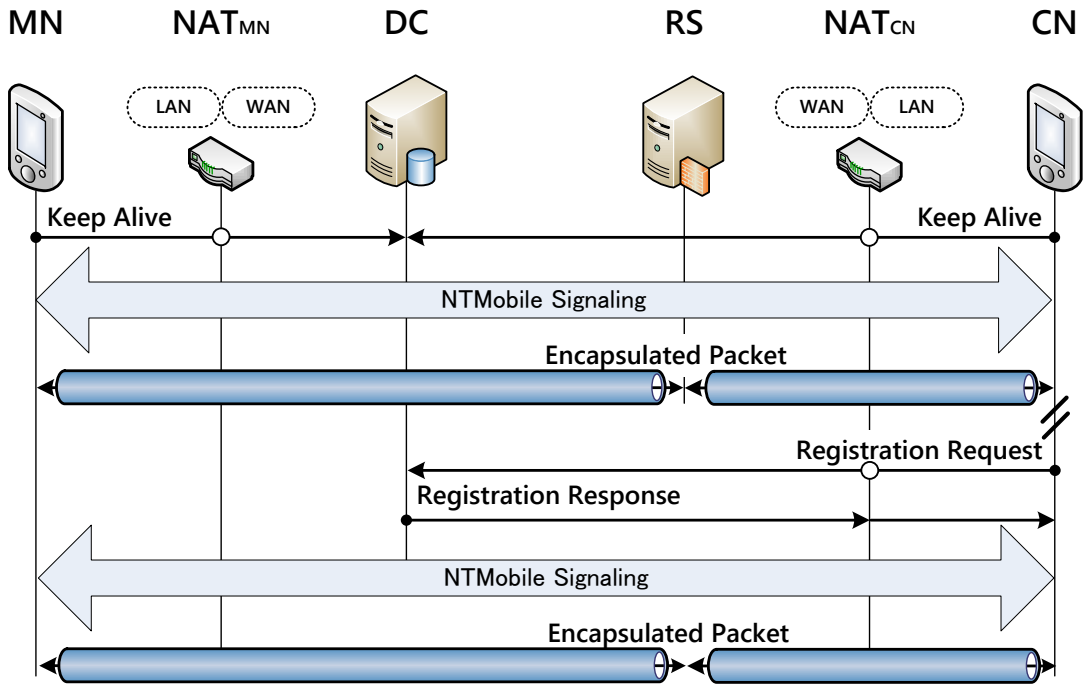

図 4　移動処理を含めたシーケンス図

リケーションを作成する段階で NTMfw を組み込む必要があるため，既存のアプリケーションには適用することが難しい．

2.6　VPNService 型 NTMobile

NTMobile をスマートデバイスアプリケーションとして実装したモデルである．実装する際に VPN 通信を行うための API（以降，VPN API）である VPNService と NTMfw を利用している．VPNService 型 NTMobile は VPN API によるカプセル化 / デカプセル化機能を利用することで，既存のアプリケーションに NTMobile 通信を実現することが可能な NTMobile のモデルである．現在，Java アプリケーションとして実装されており，C で記述された NTMfw の機能を利用するために JNA（Java Native Access）を使用している．VPNService 型 NTMobile は DWCamera や IPWebcam といったライブチャットアプリケーションに NTMobile の NAT 越えや E2E の直接通信を実現することが確認できている．しかし，NTMfw が OS によって異なるインタフェースの名前を共通で検出することが出来ないため，端末の IP アドレスが変化しても移動処理を実行することが出来ない．

第3章　プッシュ通知機能

本章では，スマートデバイス向けに提供されているプッシュ通知機能について説明する．

3.1　プッシュ通知機能概要

スマートデバイスには，アプリケーション開発者が管理するサーバからユーザの利用しているアプリケーションにプッシュ通知する機能がある．プッシュ通知機能は，任意のタイミングでサーバからアプリケーションに向けてメッセージを送信することが出来る．アプリケーションに通知を行うサーバとスマートデバイス間で実行される OS レベルでの Keep Alive を行っているため，スマートデバイスが NAT 配下に存在してもプッシュ通知を受信することが出来る．代表的なものとして Google Inc. の提供する Android 向けの GCM（Google Cloud Messaging）と GCM の後継として誕生した FCM（Firebase Cloud Messaging），Apple Inc. の提供する iOS 向けの APNs（Apple Push Notification Service）がある．FCM の機能の一つとして Firebase Cloud Messaging APNs インタフェースがあり，FCM から APNs を利用して iOS アプリに通知メッセージを送信することも可能である [9].

プッシュ通知として送信するメッセージはスマートデバイスのアプリケーションの起動トリガーとして利用することも可能であり，アプリケーション開発者が任意のタイミングでユーザがインストールしているアプリケーションを起動させることができる．

3.2　プッシュ通知機能の動作シーケンス

図 5 に例として FCM の動作シーケンスを示す．ここで，Application Server はアプリケーション開発者が管理するサーバである．FCM Server は Google Inc. の管理するサーバであり，スマートデバイスの間で行われる OS レベルの Keep Alive を行っている．この KeepAlive は TCP ベースであるため KeepAlive の間隔は UDP による KeepAlive と比較すると長い．FCM Server を利用したプッシュ通知機能を利用する際は，スマートデバイスが事前に Registration を行っておく必要がある．Registration はアプリケーションのインストール時に実行し，FCM Server にデバイス ID とアプリケーション ID の情報を登録する．スマートデバイスの Registration 完了後に FCM Server は，デバイス ID とアプリケーション ID に紐づき，一意に識別可能な Registration ID をスマートデバイスに発行する．FCM Server から Registration ID を受信したスマートデバイスは，Application Server に Registration ID を送信して登録する．

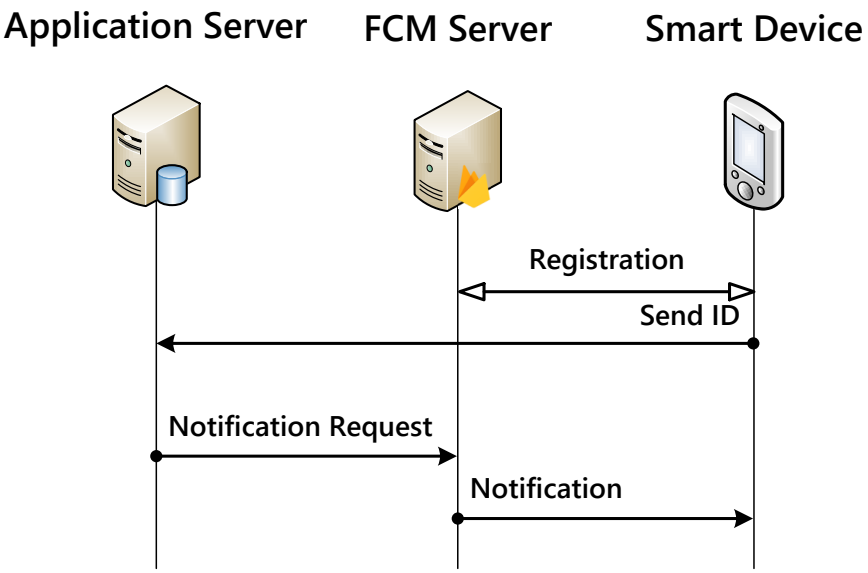

図 5　FCM の動作シーケンス

Application Server がスマートデバイスにプッシュ通知メッセージを送信する時には，FCM Server に Notification Request を送信する．Notification Request には，Registration ID と通知したいメッセージが含まれている．FCM Server は Registration ID に紐づいたスマートデバイスのアプリケーションに Notification を送信し，メッセージを通知する．

第4章　提案方式

本章では最初に，VPNService 型 NTMobile にアドレス変化検出を実現する方法について説明したのちに，プッシュ通知機能である FCM を利用した VPNService 型 NTMobile の省電力化とシグナリング処理の見直しについて述べる．

4.1　VPNService 型 NTMobile に対する移動透過性の実現

提案方式では，Android のアプリケーションからアドレス変化検出したことを NTMfw に通知することにより，端末のアドレスが変化した際に移動処理を実行させる．

4.1.1　モジュール構成

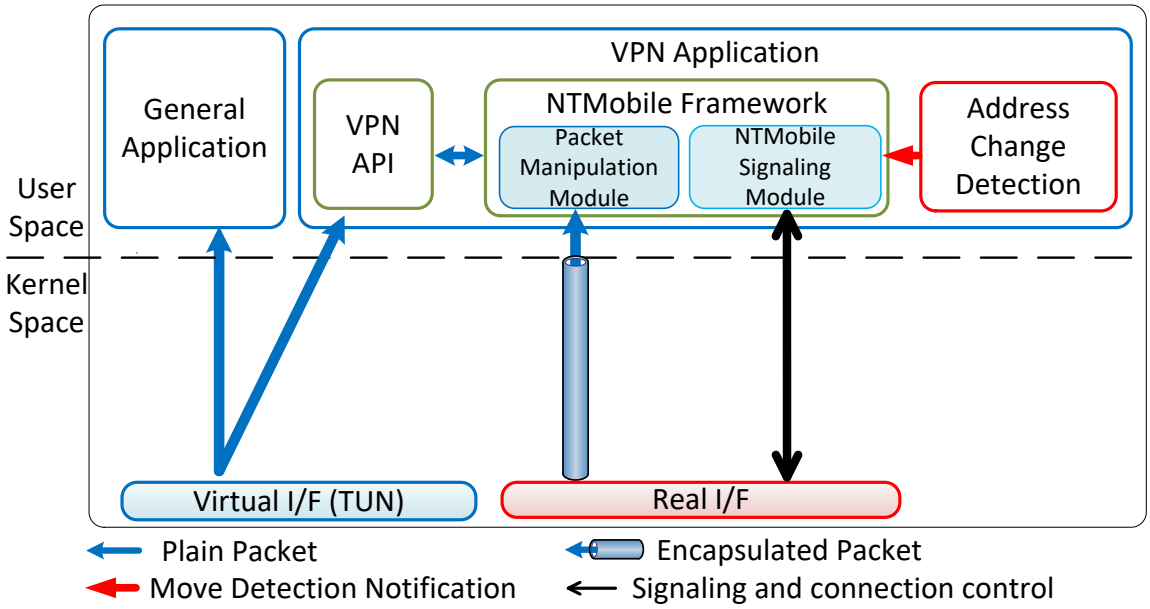

図 6　VPNService 型 NTMobile にアドレス変化検出を実装した際のモジュール図

図 6 にアドレス検出処理を独立させた VPNService 型 NTMobile のモジュール図を示す．VPNService 型 NTMobile では，まず VPN Application 起動時に VPN API が仮想インタフェースの作成を行う．その後，一般アプリケーションは仮想インタフェースを経由してパケットの送受信を行う．NTMobile Framework の中に含まれている NTMobile Signaling Module は，VPN Application 起動

時のアドレス登録処理や NTMobile によるシグナリングを行っている．また，Packet Manipulation Module では，NTMobile によるパケットの送受信処理を行っている．提案方式では，VPN Application の中にあるこれらのモジュールに追加してアドレス変化検出部分を作成し，アドレスが変化した際には NTMobile Signaling Module にアドレスが変化したことを伝える通知を送信する．その後，通知をトリガとして図 4 に示したシーケンスを実行することで VPNService 型 NTMobile に移動透過性を実現することが可能となる．

4.2　FCM を利用したシグナリング処理の見直し

スマートデバイスがモバイルデータ通信網に接続している場合には，モバイルデータ通信網で使用される NAT の特徴から，NTMobile のシグナリングを簡略化することが出来る．

4.2.1　ネットワーク構成

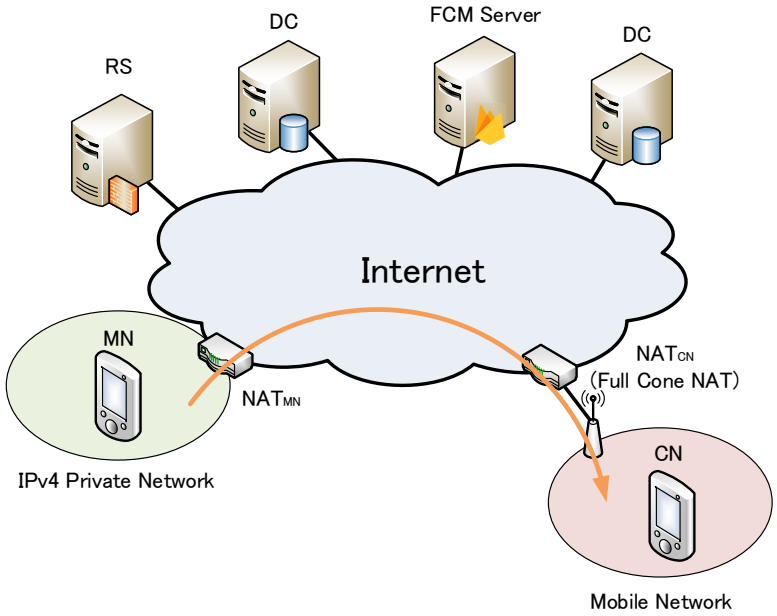

図 7　提案手法で想定しているネットワーク構成

図 7 に FCM を利用したシグナリング処理を行う上で想定しているネットワーク構成ネットワーク構成を示す．MN 側は Private IPv4 ネットワークに接続されている NTM 端末であり，CN はモバイルデータ通信網に接続されている NTM 端末である．モバイルデータ通信網で使用されている通信方式は 3G/LTE である．このときモバイルデータ通信網は巨大な Private IPv4 ネットワークとみなせる．インターネットとモバイルデータ通信網に使用される NAT_{CN} は最も制約の弱い Full Cone NAT である．FCM を利用したシグナリング処理の見直しを行う際には図 7 に示すように，MN からモバイルデータ通信網に存在する CN に対して通信を開始する場合に適用される．

4.2.2 動作シーケンス

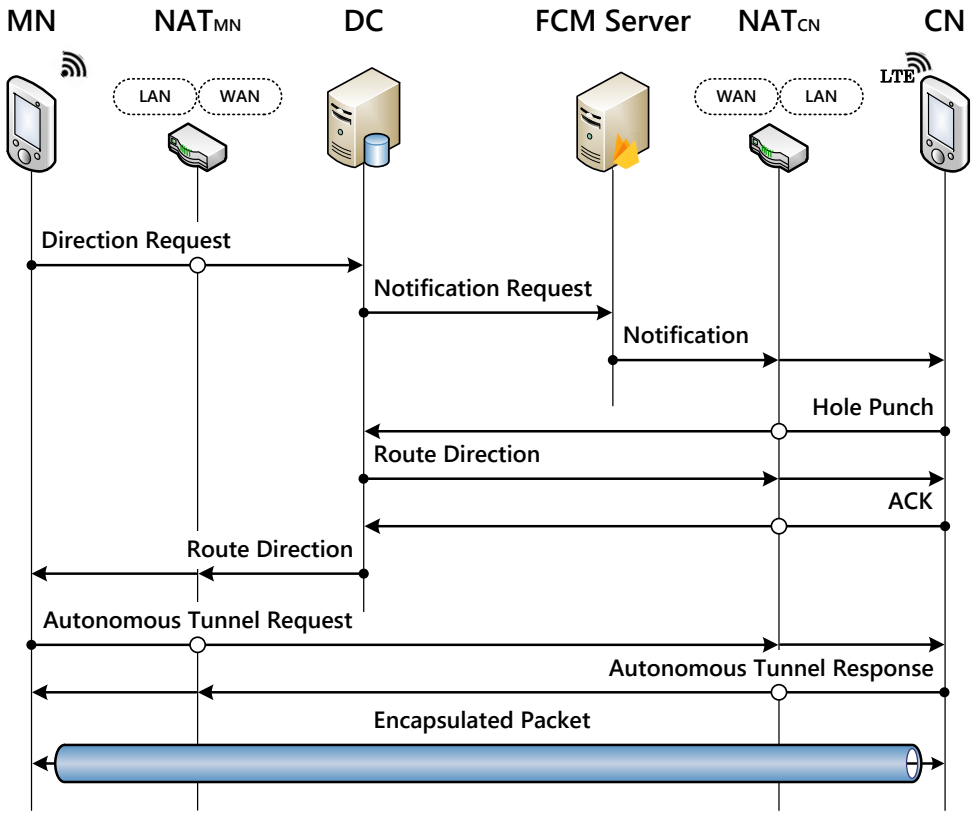

図 8　シグナリング処理の見直しを行った際のシーケンス

図 8 にシグナリング処理の見直しを行った際のシーケンスを示す．提案方式では，スマートデバイスに通知メッセージを送信するために FCM を使用し，Application Server の機能を DC に組み込む．図 8 では既にスマートデバイスが DC に端末情報の登録と FCM Server から発行された Registration ID を登録した環境を想定している．提案方式では，FCM Server とスマートデバイスの間で Keep Alive を行っているため NTMobile として DC に Keep Alive は行わなくてよい．これによって，スマートデバイスは DC に 10 秒毎に KeepAlive を送信する必要が無く，長い間隔で行われる TCP ベースの KeepAlive だけ行えばよいため，スマートデバイスに実装した NTMobile の低消費電力化が期待できる．

DC は Direction Request を受信すると DC に登録されている CN の端末情報を確認する．この時 CN がスマートデバイスであれば DC は FCM Server に Notification Request を送信する．Notification Request を受信した FCM Server は Notification Request に記述されている Registration ID の情報を基に CN に Notification を送信する．次に，CN は Notification 受信後，DC に Hole Punch を送信することで，DC からの経路指示を受信することが可能となる．DC からの指示が受けられるようになった CN は DC から Route Direction を受信した後，ACK を送信する．DC は ACK の内容から

NAT_{CN} の IP アドレスとポート番号を知ることが出来るため，DC は MN に Route Direction を送信する際に，最初から自律的経路最適化を実行するように指示することが出来る．MN は自律的経路最適化を試みるために NAT_{CN} の IP アドレスとポート番号に Autonomous Tunnel Request を送信する．この時，CN がモバイルネットワークに接続している場合には，NAT の特性から Autonomous Tunnel Request が破棄されることなく一回で CN に届くため，RS を経由せずに自律的経路最適化を試みることが出来る．

第5章　実装

本章では，提案方式の実装方法と，現段階で実装が完了しているところについて説明する．

5.1 Address Change Detection

5.1.1 実装方法

VPN Application にてアドレスの変化を検出するために，Android が提供する Connectivity Manager を使用する．[10] Connectivity Manager は，Android 端末の接続状況が変化すると，CONNECTIVITY_ACTION（“android.net.conn.CONNECTIVITY_CHANGE”）をブロードキャストする．そのため，VPN Application からアドレス変化検出を実行するためには，CONNECTIVITY_ACTION を受信するレシーバを作成する．Address Change Detection では，CONNECTIVITY_ACTION をトリガとして NTMfw の Signaling Module に通知を送信する．しかし NTMfw は C で記述されているため JNA（Java Native Access）を経由する必要がある．そのため，NTMfw に端末移動時の処理を呼び出す関数を記述した C ソースを作成後コンパイルし，共通ライブラリの作成を行った．最後に，CONNECTIVITY_ACTION を受信した際に Java 側から loadLibrary メソッドを利用し端末移動時の処理を呼び出すことで，VPN Application に移動透過性を実現する．

5.1.2 実装段階

端末移動時に実行する移動処理に関わる部分を記述し，JNA で使用するための共通ライブラリとして実装を行った．しかし，Connectivity Manager を利用した部分の作成は出来ていない．そのため，Connectivity Manager の代わりに VPN Application に ADDRESS_CHANGE ボタンを作成し，このボタンをトリガとして JNA を利用した端末移動時の処理を行った．

実際に端末移動時に正しく移動処理を実行できるかを検証するために，鈴木研究室に設置されているインターネットを模擬したローカル環境を利用して以下の実験を行った．実験を行う際，Android 向けアプリケーションとして提供されている Network Analyzer を使用した．

- MN，CN 共に NAT 配下に設置した状態で，MN から CN に定期的に Ping を送信し続ける．
- MN から送信される Ping の到達を確認した状態で，CN をグローバル空間に移動させて Ping が途切れることを確認する．
- ADDRESS_CHANGE ボタンをタップし，MN から送信される Ping が再び CN に到達することを確認する．
- CN を再び NAT 配下に移動させ，同様に移動処理が実行できるかを確認する．

結果，Android 端末が移動した際に ADDRESS_CHANGE をタップすることで，通信の継続が可能となった．

5.2　3G / LTE で使用するインタフェースの検出

NTMfw は Android 特有のインタフェースである 3G / LTE を使用して Registration 処理を実行することが出来ない．そのため，NTMfw のインタフェースの取得をする部分で，Linux 以外のインタフェースでも使用できるように書き換えを行った．そこで，Google Play ストアから一般アプリケーションとして配布されているビデオチャットアプリである DWCamera を使用して，実際に 3G / LTE を使用して NTMobile 通信が実行可能であるかを確認した．図 9-10 には，VPN Applivcation による Registration 処理を行った後の画像を提示している．VPNService 型 NTMobile では，NTMobile 通信を実現するために，VPN 接続をしている必要があるため，常にステータスバーに VPN 接続をしていることを表す鍵マークが表示される．図 11-12 は，NTMobile を利用して双方向の映像送信を行った結果をキャプチャしたものである．結果，LTE に接続している端末でも NTMobile による通信が確認できた．

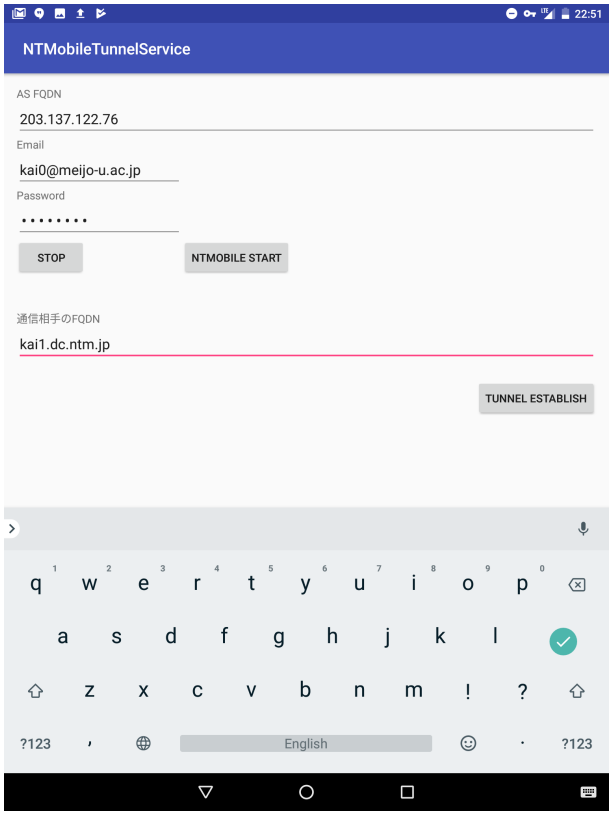

図 9　LTE 環境で接続した状態

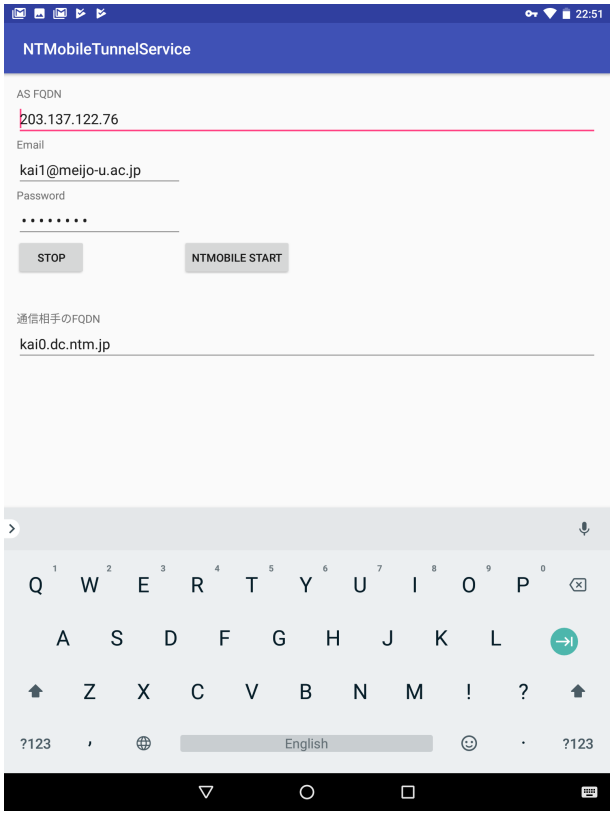

図 10　Wi-Fi 環境で接続した状態

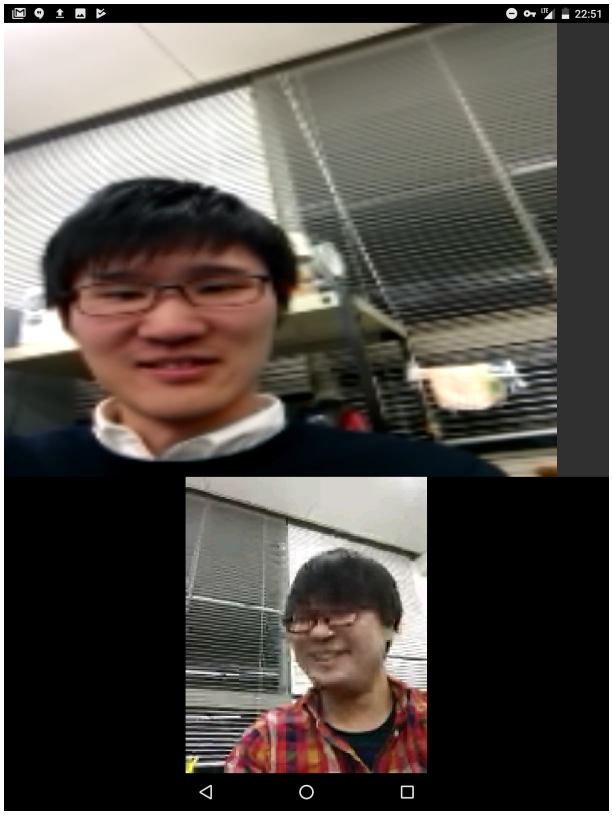

図 11　LTE 接続の端末から DWCamera で NTMobile による通信をした映像

図 12　Wi-Fi 接続の端末から DWCamera で NTMobile による通信をした映像

第6章　結論

本研究では，VPNService 型 NTMobile に移動透過性を実現するための提案と実装を行ったうえで，スマートデバイスに適した形の NTMobile を実現するために KeepAlive を削減する方式を提案し，スマートデバイス向けのシグナリング処理を再検討した．VPNService 型 NTMobile に移動透過性を実現する方式では，NTMfw が端末のアドレスが変化した際の検出方法と，アドレスの更新処理について提案と実装を行い，端末移動時に NTMobile 通信の継続を確認した．

しかし，Connectivity Manager を利用したアドレス変化のトリガ部分と，KeepAlive の削減する方式，シグナリング処理の再検討部分に関しては実装が出来ていないため，今後実装を行った上で，評価を行う．

謝辞

本研究を進めるにあたり，多大なるご指導とご教授を賜りました，名城大学理工学部情報工学科の渡邊晃教授に心から感謝いたします.

本研究を進めるにあたり，様々なご指導とご意見を賜りました，名城大学理工学部情報工学科の鈴木秀和准教授に深く感謝いたします.

本研究を進めるにあたり，ご意見とご助言を賜りました，愛知工業大学情報科学部の内藤克浩准教授に深く感謝いたします.

最後に，本研究を進めるにあたり，親身かつ丁寧なご指導を賜りました，渡邊研究室及び鈴木研究室の先輩方と，数々の有益なご助言を賜りました渡邊研究室及び鈴木研究室の同期の皆様に感謝いたします.

参考文献

[1] : Cisco Virtual Networking Index : 全世界のモバイルデータトラフィックの予測、2015 〜 2020 年アップデート. https://www.cisco.com/c/ja_jp/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html.

[2] 鈴木秀和，水谷智大，西尾拓也，内藤克浩，渡邊　晃：NTMobile における相互接続性の確立手法と実装，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. 1339–1348 (2011).

[3] 鈴木秀和，上酔尾一真，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] 内藤克浩，上酔尾一真，水谷智大，西尾拓也，鈴木秀和，渡邊　晃：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).

[5] 上酔尾一真，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. 52, No. 9, pp. 2549–2561 (2011).

[6] 山田貴之，鈴木秀和，内藤克浩，渡邊　晃：IPv4/IPv6 混在環境に対応した VPNService 型 NTMobile の性能評価，マルチメディア，分散，協調とモバイル（DICOMO2015）シンポジウム論文集，Vol. 2015, pp. 1792–1799 (2015).

[7] : Firebase Cloud Messaging - Google. https://firebase.google.com/?hl=ja.

[8] 納堂博史，鈴木秀和，内藤 克浩晃：NTMobile における自律的経路最適化の提案，情報処理学会論文誌，Vol. 54, No. 1, pp. 394–403 (2013).

[9] : Google Developers Japan: iOS で Firebase Cloud Messaging をデバッグする. https://developers-jp.googleblog.com/2017/02/debugging-firebase-cloud-messaging-on.html.

[10] : ConnectivityManager — Android Developers. https://developer.android.com/reference/android/net/ConnectivityManager.html.

研究業績

研究会・大会等（査読なし）

（1）黒宮魁人，清水一輝，鈴木秀和，内藤克浩，渡邊 晃：スマートデバイスにおける NTMobile の経路生成方式の提案，平成 29 年度電気・電子・情報関係学会東海支部連合大会論文集，Vol.2017, No.C3-4, Sep. 2017.

（2）黒宮魁人，清水一輝，鈴木秀和，内藤克浩，渡邊 晃：スマートデバイスにおける NTMobile の経路生成方式の提案，第 15 回情報学ワークショップ（WiNF2017）論文集，Vol.2017, No.D-4, Nov. 2017.

（3）黒宮魁人，清水一輝，鈴木秀和，内藤克浩，渡邊 晃：VPNService を利用した移動透過性実現方式の提案，第 80 回情報処理学会全国大会講演論文集，Vol.2017, No.1，6T-07, Mar. 2018.