Vol.2020-MBL-97 No.6
Vol.2020-ITS-83 No.6
2020/11/24

拡張NTMobileによるIP Flow Mobilityの Androidスマートフォン上における基礎的検証

松岡　穂[1,a)]　鈴木　秀和[1,b)]　内藤　克浩[2]

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．しかし，ユーザの移動や選択による接続先ネットワークの切り替えにより全てのトラフィックがオフロードされ，移動先ネットワークの通信状況・品質によっては QoE（Quality of Experience）の低下が懸念される．この課題の解決策として 3GPP（3rd Generation Partnership Project）で議論されている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 が挙げられる．筆者らは移動透過性技術である NTMobile（Network Traversal with Mobility）をユーザ空間で拡張することでカーネル空間の改造を不要とした通信断絶の無い IP Flow Mobility を実現する手法を提案してきた．本稿では，Android スマートフォンおよび NTMobile サーバに拡張実装を行い，基礎的検証として NTMobile トンネル追加構築処理オーバヘッドを確認する．

A Basic Verification of IP Flow Mobility for AndroidSmartphone by Extended NTMobile

Matsuoka Minoru[1,a)]　Suzuki Hidekazu[1,b)]　Naito Katsuhiro[2]

1. はじめに

スマートフォンやタブレットなどのモバイル端末の普及に伴い，モバイルデータトラフィックが増加しており，2022 年には毎月約 77.5 エクサバイトの通信量が発生することが予測されている [1]．このデータトラフィック増加の対応策の 1 つとして，Wi-Fi オフローディングが挙げられ，現在ユーザは外出時はモバイルネットワークを利用して通信を行い，自宅や会社にいる場合は Wi-Fi ネットワークを利用するなど，モバイルと Wi-Fi を切り替えながら通信を行う機会が増加している．しかし，モバイル端末ユーザの移動や選択による接続先ネットワークの切り替えにより全てのトラフィックがオフロードされ，移動先ネットワークの通信状況・品質によっては VoIP といったリアルタイム性を要求する通信フローなどの QoE（Quality of Experience）の低下が懸念される．また，現在第 5 世代通信システム（5G）の導入・サービスインが進んでおり [2]，5G ネットワークを介して超低遅延通信を行っていたユーザが Wi-Fi ネットワークにハンドオーバした場合，通信品質の低下が予想される．

この懸念される課題を解決する技術の 1 つとして 3GPP（3rd Generation Partnership Project）で議論されているモバイルと Wi-Fi を同時利用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 が挙げられる [3]．IP Flow Mobility には大きく分けてネットワークベースアプローチとクライアントベースの 2 種類のアプローチがあり [4]，筆者らはクライアントベースのアプローチとして，IPv4/IPv6 ネットワークにおいて IP モビリティとエンドツーエンド接続性の両方を提供する NTMobile（Network Traversal with Mobility）[5], [6], [7] をユーザ空間で拡張することにより，カーネル空間の改造を不要とし

[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)] minoru.matsuoka@ucl.meijo-u.ac.jp
[b)] hsuzuki@meijo-u.ac.jp

た通信断絶の無い IP Flow Mobility を実現する手法を提案してきた [8]．提案手法では，現在モバイルネットワークは LTE をターゲットとしており，LTE と Wi-Fi の両方に接続した端末が一定間隔で測定した LTE と Wi-Fi の RSSI（Recieved Signal Strength Indicator）および RTT（Round Trip Time）をトリガとしてハンドオーバを事前に検知し，通信相手への追加トンネル構築を行う．トンネルの構築完了後，通信フローに対してどの通信インタフェースを利用するか記載されたルーティングポリシーに基づき，通信フローを各トンネルに対して振り分ける．

本稿では，Android スマートフォン上で動作可能な NTMobile に対してユーザ空間で拡張実装を行い，また NTMobile トンネル経路指示を行う NTMobile サーバに対しても拡張実装を行う．基礎的検証として，AWS EC2（Amazon Web Service Elastic Computer Cloud）[9] 上に各 NTMobile サーバを構築し，LTE と Wi-Fi の両方に接続した Android スマートフォン上で拡張 NTMobile を動作させ，NTMobile トンネル追加構築処理に関するオーバヘッドを確認する．

以下，2 章で関連研究とその課題，3 章で提案手法の動作および必要となる拡張機能について述べ，4 章で基礎的実装および検証結果について示し，最後に 5 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

2.1.1 PMIPv6 を拡張した手法

T. Melia らは，PMIPv6（Proxy Mobile IPv6）における LMA（Local Mobility Anchor）と MAG（Mobile Access Gateway）の機能を拡張することで IP Flow Mobility を実現している [10]．LMA は，フロー割当先の端末 ID，フロー ID，フロー割当先 MAG の組み合わせを管理する Flow Binding Cache を新たに作成し，どのネットワークに対しフローを割り当てるか決定する主体となる．MAG は，LMA のフロー割当決定に応じて管理するルーティング状態を更新し，該当フローを転送するよう拡張される．これらの変更により，拡張 PMIPv6 では IP フローモビリティが可能となる．

しかし，端末が移動して 1 つの MAG のみに接続が変更された場合，LMA は接続を失ってから以前接続していた MAG に振り分けていたフローを移動させるため，パケットロスが発生し，QoE の低下が懸念される．

2.1.2 PMIPv6 と SDN を利用した手法

D. Purohith らは，PMIPv6 と SDN（Software Defined Networking）を組み合わせた SIFM（Seamless Internetworking Flow Mobility）アーキテクチャにより IP Flow Mobility を実現している [11]．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）で構成される．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 の間にはトンネルが事前に構築される．FC によるフロー割当て指示に基づいて，PGW および WAG は動作し，フローは PGW と WAG 間のトンネルを経由し 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．

しかし，T. Melia らの手法と同様に，モバイル端末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 FC によりフロー割当変更がなされるため，パケットロスが発生し，QoE の低下が懸念される．

2.2 クライアント主導型 IP Flow Mobility

2.2.1 Mobile IPv6 を拡張した手法

N. Varga らは，HoA（Home of Address）に対して複数の CoA（Care of Address）登録をサポートする RFC5648[12] と通信フローと HoA，CoA の組み合わせを対応づけることでフロー振り分けを可能にする RFC6088[13] に基づいて拡張したカーネルモジュール（MIP6D-NG）と Android アプリケーションをクロスレイヤで連携させることで IP Flow Mobility を実現している [14]．Android アプリケーションは，LTE に加えて Wi-Fi に追加接続してマルチホームとなった場合，Wi-Fi ネットワークの RSSI，RTT，ジッタ，パケットロスを測定し，測定値が適切であった場合，QoS プロファイルに基づいて該当フローを Wi-Fi ネットワークに移動させる決定を下す．決定を受けた MIP6D-NG カーネルモジュールは，HA（Home Agent）に Flow Binding Update（FBU）を送信し，HA は適切なフロールーティング情報を生成し，HA はモバイル端末に対して Flow Binding Ack を返す．この手順により，IP Flow Mobility が可能となる．

しかし，この手法ではモバイル端末のカーネル空間の改造が必要となり，一般ユーザの利用のハードルが高いという課題が存在する．

3. 提案手法

3.1 概要

本研究では筆者らがこれまで提案してきた NTMobile の仕様を拡張することにより，IP Flow Mobility を実現する．**図 1** に提案手法の概要を示す．ユーザが所有するモバイル端末である UE（User Equipment）およびその通信相手である CN（Correspondent Node）は拡張 NTMobile が実装された NTM 端末である．NTM 端末は拡張 DC（Direction Coordinator）より割り当てられたネットワークにおいて不変である仮想 IP アドレスをアプリケーションに認識させてトンネル通信を行い，実 IP アドレスの変化を隠ぺい

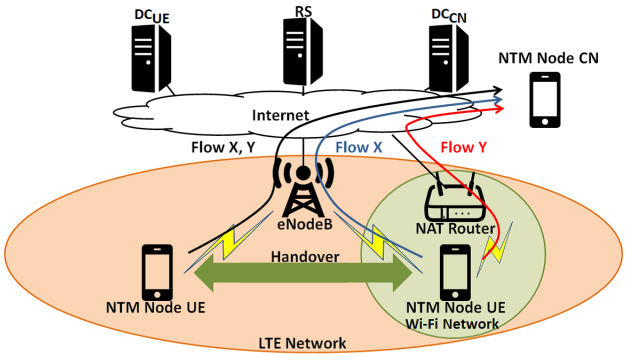

図 1 NTMobile における IP Flow Mobility の概要

表 1 想定する提案手法における通信パターン

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

する．

拡張 DC_N は NTM 端末 N が通信に用いる仮想 IP アドレス割り当ておよび各種アドレス情報管理，トンネル経路指示を行う．拡張 DC は NTM 端末の複数のアドレス情報を管理可能であるように拡張され，UE および CN に対して複数のトンネル経路指示を行うように拡張される．

RS（Relay Server）は異なる NAT 配下のプライベートネットワークに属する NTM 端末間の通信や IPv4/IPv6 ネットワーク間通信など直接通信ができない場合にパケットを中継する．

UE および CN は LTE 用，Wi-Fi 用のルーティングテーブルを事前に設定しているものとする．UE から CN へのフローを Flow X，Y とする．

提案手法では，NTM 端末が LTE もしくは Wi-Fi のシングルホームから LTE と Wi-Fi の両方に接続したマルチホーム状態に推移した場合，移動先ネットワークの通信状況を確認した後ハンドオーバを実行し追加トンネル構築処理を行う．マルチホームからシングルホームへの切り替えに関しては従来の NTMobile によるシグナリングを要することなく，通信断絶前にハンドオーバを実行する．

3.2 ハンドオーバシーケンス

提案手法では，**表 1** に示す通信パターンを想定しており，ここではパターン 1 に該当する，シングルホーム接続の CN に対して，LTE で通信を行っていた NTM 端末が新たに Wi-Fi に追加接続してマルチホーム状態に移行した場合の通信シーケンスについて述べる（**図 2**）．また，シングルホーム接続の CN に対して，マルチホーム状態で複数のトンネルを使用して通信を行っていた UE が移動して LTE のみのシングルホーム接続となった場合のパターン 2 における通信シーケンスについて述べる（**図 3**）．

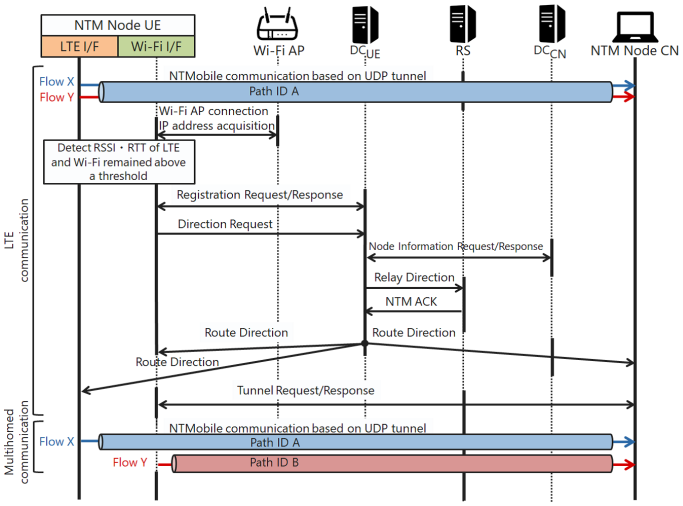

図 2 提案手法におけるハンドオーバシーケンス，パターン 1（CN: シングルホーム，UE: LTE からマルチホーム）

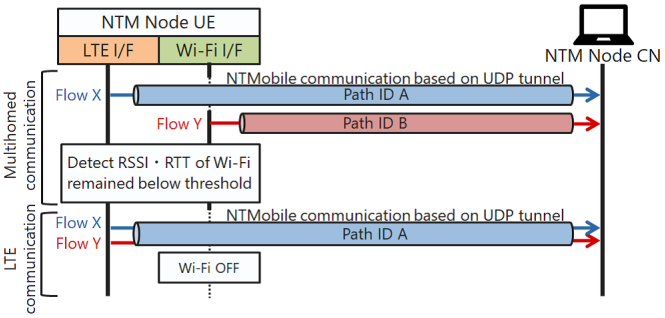

図 3 提案手法におけるハンドオーバシーケンス，パターン 2（CN: シングルホーム，UE: マルチホームから LTE）

3.2.1 追加ネットワーク接続および追加アドレス登録

UE 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 NTMobile トンネル通信を行っているものとする．UE が移動して LTE に加えて Wi-Fi に追加接続した場合，UE は移動先ネットワーク状況確認として LTE と Wi-Fi の RSSI および RTT を一定間隔で測定する．それらが一定時間の間閾値以上であり続けた場合，UE は移動先ネットワークである Wi-Fi を利用して DC_{UE} 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．拡張 Registration Request は従来の Registration Request に対して新たに拡張データとして通信インタフェース名（Wi-Fi であれば wlan0）を追加したメッセージである．

3.2.2 追加トンネル構築

アドレス登録を完了した UE は，DC_{UE} 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_{UE} は DC_{CN} と Node Info-

marion Request/Response を行い CN のアドレス情報を取得する．なお，DC1 台が UE および CN のアドレス情報を管理していた場合，このやり取りは発生しない．その後，DC_{UE} は UE および CN（DC_{CN} 経由）へ複数の拡張 Route Direction を送信し，トンネル構築経路指示を行う．DC_{UE} は UE 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で，考えられる全ての通りのトンネル構築経路指示を行う．拡張 Route Direction は従来の Route Direction に対して新たに拡張データとして自身の通信インタフェース名と通信相手の通信インタフェース名，トンネル構築数を追加したメッセージである．経路指示を受けた UE および CN は複数の Tunnel Request/Response をやり取りし，トンネル追加構築処理を完了する．

3.2.3 フロー振り分け

トンネル追加構築を完了した UE 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．UE は通信フローに対してどの通信インタフェースを利用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 UE から見て LTE 側のトンネルに，Flow Y を UE から見て Wi-Fi 側のトンネルに振り分けたものとする．

シングルホーム接続の CN に対して，UE がマルチホームから Wi-Fi にハンドオーバした場合の通信シーケンスについて述べる．Wi-Fi の RSSI と RTT が一定時間の間閾値より小さくあり続けたことを検知した UE は UE から見て LTE 側のトンネルキープアライブを利用して CN に対して UE 自身の Wi-Fi が接続終了することを通知する．また UE は DC_{UE} に対して，拡張 Registration Request を送信し，Wi-Fi の接続終了を通知し，DC_{UE} はそれを受けて管理していた UE の Wi-Fi アドレス情報を削除する．UE は通知後，全ての通信フローを LTE トンネルに割り当て，通信断絶前にハンドオーバを完了する．

これにより，従来の NTMobile のようなハンドオーバ用のトンネル構築シーケンス（Registration Request，Route Direction，Tunnel Request/Response）を排した即時切替が実現される．

3.3 NTMobile の拡張

3.3.1 拡張ヘッダおよび拡張データの利用

NTMobile では実 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または仮想 IP パケット，MAC（Message Authentication Code）の順でパケットが構成されている．拡張ヘッダを利用して拡張データを追加することで従来の NTMobile メッセージを拡張することが可能となる．拡張データフォーマットは**図 5** の通りである．

DC が NTM 端末の複数のアドレス情報を管理するため

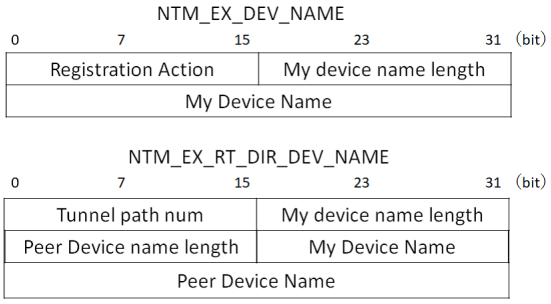

図 4 拡張データフォーマット

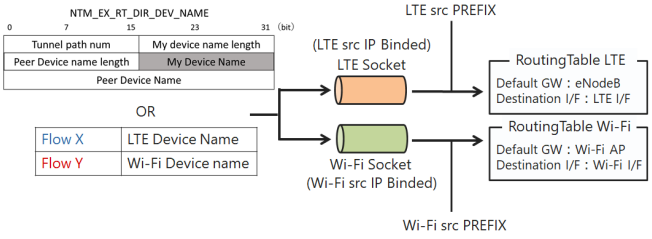

図 5 ソケットスイッチング

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）を付与することでアドレス登録を行うよう拡張する．また，DC が NTM 端末に対して複数の Route Direction を送信してトンネル経路指示を行う際，NTM 端末自身の通信インタフェース名と通信相手の通信インタフェース名，トンネル構築数を記載した拡張データ（NTM_EX_RT_DIR_DEV_NAME）を付与することで，NTM 端末の通信における使用インタフェースの切り替えおよび構築数に応じた複数トンネル構築に関わるスレッド立ち上げを行うことができるよう拡張する．

3.3.2 NTM 端末の拡張

NTM 端末は拡張データの付与および拡張データの取得ができるよう拡張される．また，NTM 端末は通信インタフェースを指定することでそれに関連付けられたソケットを使用し，使用するトンネルのスイッチングを実現する．ソケットを切り替えるパターンとして，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 用のソケットは各送信元 IP アドレスでバインドされ，また送信元 IP アドレスプレフィックスにより該当するルーティングテーブルが参照され，各ネットワークへとパケットが送信される（図）．

マルチホーム時に NTM 端末は接続ネットワークの通信品質確認のために LTE と Wi-Fi の両方で RSSI を取得し，また RTT は自身を管理する DC に対して LTE と Wi-Fi の

表 2　拡張 node_address テーブル

id	node_id	dev_name	rip4	rip6	nat_port
1	nodeA	wlan0	rip4_wifi	rip6_wifi	portA
2	nodeA	rmnet_usb2	rip4_lte	rip6_lte	portB

両方で取得し，ハンドオーバ処理を行うように拡張される．

3.3.3　DC の拡張

DC は拡張データの付与および拡張データの取得ができるよう拡張される．また，DC は拡張 Registration Request を受信した際，アドレス情報を管理する node_address テーブルに対して保存を行うが，従来の node_address テーブルに対して新たに dev_name フィールドを追加することで，通信インタフェース名を保存し，NTM 端末の複数アドレス情報を管理できるよう拡張される．拡張 DC における node_address テーブルは**表 2** の通りである．なお，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して示されている．

4.　基礎的実装および検証

4.1　基礎的実装

NTM 端末である Android スマートフォンおよび DC に対して基礎的実装を行った．Android スマートフォン上で動作する拡張 NTMobile アプリケーションは Android フレームワークの VpnService を利用した NTMobile[15] をユーザ空間で拡張したものである．RSSI および RTT の取得，ハンドオーバ判断機能，送信元プレフィックスによるルーティングテーブル参照機能は Android フレームワークを利用し，Kotlin により実装した．

ルーティングテーブル参照機能はアプリケーション内部で Android に標準装備されている `iproute2` による `ip rule` コマンドを実行することで実現される．これは root 権限を必要とするが，マルチホーム通信の実現上現状では回避不可能である．なお，もう 1 つの方法として `SO_BINDTODEVICE` オプションを利用した `setsockopt` 関数による方法もあるがこれも root 権限が必要である．

JNA（Java Native Access）経由で呼び出される NTMobile ライブラリは C 言語で実装し，拡張データの付与・取得，通信インタフェース名によるソケットスイッチング機能を追加した．DC は拡張データの付与・取得機能および複数トンネル構築指示機能を C 言語で実装した．

4.2　基礎的検証

4.3　検証環境

図 6 に基礎的検証を行った環境を示す．各 NTMobile サーバ DC1 台，RS1 台は AWS EC2 上の東京リージョンに設置した．また各サーバの EC2 インスタンス諸元は**表 3** の通りである．UE は Android スマートフォン（Pixel3，Android 9.0）を，CN は Android スマートフォン（Pixel3a，

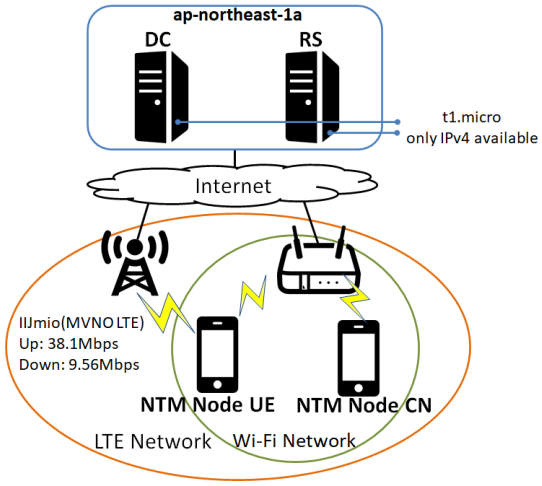

図 6　基礎的検証環境

表 3　各サーバの AWS EC2 インスタンス諸元

	DC, RS
Availability Zone	ap-notheast-1a
Instance Type	t1.micro
OS	Ubuntu 14.04 LTS 32bit
vCPU	1
Memory	0.613 GiB

表 4　各端末間の RTT

	Min [ms]	Avg. [ms]	Max [ms]
UE_{LTE} − DC	32.44	46.93	320.82
UE_{LTE} − RS	31.61	44.06	97.30
UE_{WIFI} − DC	18.87	21.58	24.47
UE_{WIFI} − RS	17.79	21.83	129.12
CN_{WIFI} − DC	22.65	36.91	75.47
CN_{WIFI} − RS	28.55	35.90	46.78
RS − DC	0.36	0.46	0.84

Android 9.0）を使用した．UE は IIJmio による MVNO LTE 回線（docomo）および Wi-Fi 回線を使用し，CN は UE と同一 NAT 配下の Wi-Fi 回線を使用する．

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．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 回送受信した．**表 4** に各端末間の RTT を示す．なお，検証時は平日 16 時 30 分頃である．

4.4　検証シナリオ

UE は LTE に加えて Wi-Fi ネットワークに接続しマルチホーム状態になっているものとする．また，UE は既に DC への LTE アドレス情報登録を完了しているものとする．DC は常に Wi-Fi 接続のみのシングルホーム状態であり，既に 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．UE 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 の受信完了したトンネル追加構築完了までの処理時間を 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通りである．

(1) UE が LTE に加えて Wi-Fi に追加接続し，100ms 秒間隔で 3 回 LTE と Wi-Fi の両方で DC への RTT 値および RSSI を取得し，トンネル追加構築を行う判断を下す（今回は必ず追加構築する判断を下すよう設定）．
(2) UE が DC に対して拡張 Registration Request を Wi-Fi 経由で送信
(3) UE が DC から Registration Response を Wi-Fi 経由で受信
(4) UE が Direction Request を Wi-Fi 経由で送信
(5) UE が Route Direction を LTE，Wi-Fi 経由で各 1 回受信
(6) UE 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) UE が RS 経由で CN から LTE，Wi-Fi 経由で各 1 回 Tunnel Request を受信してトンネル追加構築完了

4.5 検証結果と考察

UE におけるハンドオーバ検知開始から追加トンネル構築までの各処理時間を**表 5** に示す．ハンドオーバ検知に関して，一度取得した RTT 値を考慮して測定間隔を調整し，100ms 以内の RTT 値であれば常に測定間隔が 100ms となるように実装されている．しかし，UE が利用している MVNO LTE 回線では RTT 値が刻一刻と大幅に上下動するため，1 回目のハンドオーバ判断を下すまでに 300ms 以上の時間がかかる結果となった．

拡張 Registration Request/Response に関して，平均でも 662.25ms であり，最も処理時間がかかる結果となった．この処理時間は最小 422.06ms，最大 995.70ms となり，573.64ms 秒の差が存在する．UE_{WIFI} と DC 間の RTT 値は良好であり，この差が発生した要因として DC 自体の処理スペックが 1 つの要因として考えられる．

Direction Request 送信及び複数の Tunnel Response 受信に関して，平均 506ms の処理時間がかかる結果となった．実装として，DC から CN，UE への 1 つめの拡張 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 が並行処理されることになる．これにより実装上オーバヘッドを最小に抑えられるが，MVNO LTE 回線による RTT により処理時間が増大していると考えられる．

ハンドオーバ検知開始から追加トンネル構築完了までの平均処理時間は 1588.23ms である．しかしながら，トンネルの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るため動作上問題なく，QoE の維持および向上を狙うことが可能である．

5. まとめ

本稿では IP Flow Mobility を実現するために NTMobile を拡張し，Android スマートフォン上および NTMobile サーバである DC に実装した．市販の Android スマートフォンと AWS EC2 上に構築した NTMobile サーバを利用して基礎的動作検証を行った．その結果，Android スマートフォンをマルチホーム状態とし，LTE 側と Wi-Fi 側の双方でトンネルを構築できることを確認した．また，同環境にてハンドオーバ時におけるトンネル追加構築が完了するまでの処理時間を測定した結果，約 1.6 秒で完了することを確認した．

今後，実装をさらに進め，通信断絶のない各トンネルへのフロー割当を確認，評価する予定である．

表 5 UE におけるハンドオーバ検知開始から追加トンネル構築までの各処理時間

	Min [ms]	Avg. [ms]	Max [ms]
Detect handover	406.87	455.12	510.13
ExRegistration Request ～Registration Response	422.06	622.25	995.70
Direction Request ～Multi Tunnel Response	460.56	506.07	553.93
JNA processing	1.71	4.79	15.73

参考文献

[1] Cisco: Cisco Visual Networking Index: Forecast and Trends, 2017-2022 (2018). https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.pdf.
[2] : 総務省：令和 2 年版情報通信白書，pp. 31–32.
[3] G.Giaretta: IP flow mobility and seamless Wireless Local Area Network (WLAN) offload; Stage 2, TS 23.261, 3GPP (2018).
[4] 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).
[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, No. 1, 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).
[7] 上醉尾一真，鈴木秀和，内藤克浩，渡邊晃：IPv4/IPv6 混在環境で移動透過性を実現する NTMobile の実装と評価，情報処理学会論文誌，Vol. 54, No. 10, pp. 2288–2299 (2013).
[8] Matsuoka, M., Yanase, T., Suzuki, H., Watanabe, A. and Naito, K.: An Examination of IP Flow Mobility for Android Smartphone by Extended NTMobile, *2019 IEEE 8th Global Conference on Consumer Electronics (GCCE)*, pp. 283–284 (2019).
[9] : Amazon Web Services：Amazon EC2（安全でスケーラブルなクラウド上の仮想サーバー）. https://aws.amazon.com/jp/ec2/.
[10] 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).
[11] 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).
[12] Wakikawa, R., Devarapalli, V., Tsirtsis, G., Ernst, T. and Nagami, K.: Multiple Care-of Addresses Registration, RFC 5648, IETF (2009).
[13] Tsirtsis, G., Giarreta, G., Soliman, H. and Montavont, N.: Traffic Selectors for Flow Bindings, RFC 6088, IETF (2011).
[14] Varga, N., LászlóBokor, Bouroz, S., Lecroart, B., AndrásTakács : 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), DOI: https://doi.org/10.1016/j.procs.2014.10.028 (2014). Fourth International Conference on Selected Topics in Mobile & Wireless Networking (MoWNet' 2014).
[15] Takayuki, Y., Hidekazu, S., Katsuhiro, N. and Akira, W.: 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).