

MPTCP を用いた Wi-Fi ハンドオーバーの検証 Verification of Wi-Fi Hand-over with MPTCP

土屋 勇人[†] 堤下 裕次郎[†] 鄭 俊俊[†] 齋藤 彰一[‡] 毛利 公一[†] 瀧本 栄二[†]
Hayato Tsuchiya Yujiro Teika Junjun Zheng Shoichi Saito Koichi Mouri Eiji Takimoto

1. はじめに

2020 年に開催される東京オリンピックに向けて Free Wi-Fi の整備が進んでいる。Wi-Fi 通信の欠点として、ユーザの移動によるアクセスポイントの変更により、TCP コネクションが切断されることが挙げられる。一度 TCP コネクションが切断されると、再度コネクションを確立する必要がある。このとき、ウィンドウサイズ等が初期化され、かつアプリケーションによっては通信失敗によるリトライを 1 から行うため、非効率である。この課題に対して、Mobile IP ではホームエージェントと呼ばれる第 3 の端末を用いることで解決している。しかし、コネクションは維持できるものの、ハンドオーバーによって発生する一時的に通信が途絶するなどの課題もある。

そこで、本研究では、複数 Wi-Fi インタフェースを搭載したモバイル端末上での Multipath-TCP (MPTCP) を用いた Wi-Fi ハンドオーバーにおけるコネクション維持を検討する。MPTCP は通信インタフェースごとに同一通信先と TCP コネクションを確立する。これにより、ハンドオーバーによって一方のコネクションが切断されたとしても、もう一方のコネクションを用いて MPTCP としてのコネクションを維持することができる。本研究では、この特徴を利用してハンドオーバーを交互に行うことでコネクションを維持したままでハンドオーバーを可能とする。これを実現するため、各 Wi-Fi インタフェースを可能な限り異なるアクセスポイントに接続し、MPTCP コネクションを確立する。受信電波強度が弱いアクセスポイントに接続した Wi-Fi インタフェースについては、定期的に周辺アクセスポイントを探索する。探索の結果、より受信電波強度が強いアクセスポイントを発見すると、当該アクセスポイントにハンドオーバーし、再度コネクションを確立する。この動作を繰り返すことで、上記目的を達成する。

以下、本稿では、提案手法の詳細とそのプロトタイプ実装について述べる。また、プロトタイプ実装を用いた動作検証とハンドオーバーによる通信性能への影響について考察する。

2. 関連研究

コネクションを途切れないようにハンドオーバーを行う手法として、Mobile IP[2]とパケットロスレスハンドオーバー[3]がある。

Mobile IP は、ユーザの端末がアクセスポイントを移動した際、移動前と移動後に通信する端末が同じであることを

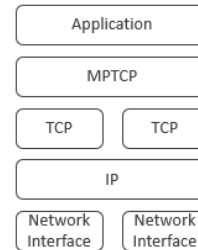


図 1 MPTCP の基本構成

サーバ側に示すために、2 つの IP アドレスを用いる。2 つの IP アドレスとは、各端末が持つ固有のホームアドレスと、アクセスポイントごとに変わる気付きアドレスの 2 つである。これらのアドレスをホームエージェントに登録しておく。ホームエージェントとは、ユーザの端末が外部のネットワークに存在している場合に、ユーザの端末宛にきたパケットを正確に現在ユーザの端末がいる場所に送信する役割を持つ。ユーザの端末が移動をすると、移動先からホームエージェント経由でユーザの端末へと通信が送られるため、送信元の把握が曖昧になってしまう。このとき、ユーザの端末は自ら気付きアドレスを取得する。送られてきたデータは、ホームエージェントとユーザの端末との間で IP トンネリングを作成して転送している。ただし、この手法では移動するネットワークごとに DHCP にてアドレスが割り当てられてしまうため、アドレスのリソースを浪費してしまう。また、パケットの送信時は端末から直接サーバに送信するが、送信元 IP アドレスがホームアドレスとなるため、受信時にはホームエージェント経由となってしまう、提案手法と比べて通信経路が長くなっている。また、Mobile IP ではネットワークレイヤでのアプローチとなる。

パケットロスレスハンドオーバーは、エンドツーエンドの端末のみの改変でハンドオーバーを行う点において提案手法と同じである。この手法は、IP アドレスの変換テーブルと Mobile PPC[4]を用いてハンドオーバーによる IP アドレスの変更をトランスポート層から隠蔽する。これにより、ハンドオーバーに伴う TCP コネクション切断を回避する。したがって、トランスポート層でハンドオーバーに対処する MPTCP と異なる階層での対策である。また、MPTCP でのハンドオーバーと違い、通信相手先に、新しいネットワークインタフェースが前に通信を行っていたものと同一であることを事前に知らせることができないため、Mobile PPC に間に挟む必要が生じる。こちらも Mobile IP と同じく、ネットワークレイヤでのアプローチとなる。

3. MPTCP の概要

MPTCP は、複数のインタフェースで複数の IP アドレスを同時に使用して通信をすることである通信の性能を向上

[†] 立命館大学 Ritsumeikan University

[‡] 名古屋工業大学 Nagoya Institute of Technology University

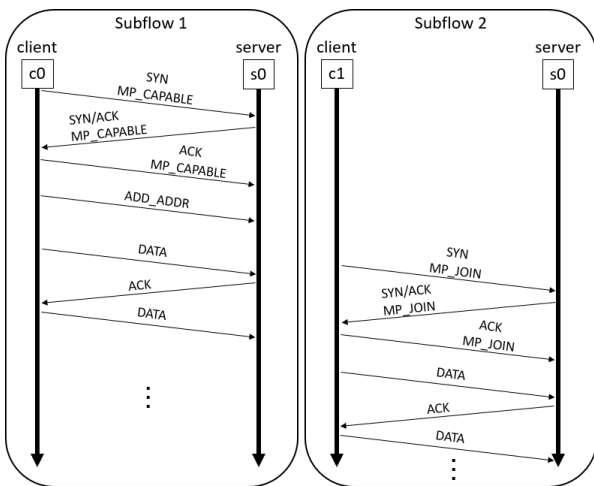


図 2 MPTCP の通信確立の流れ

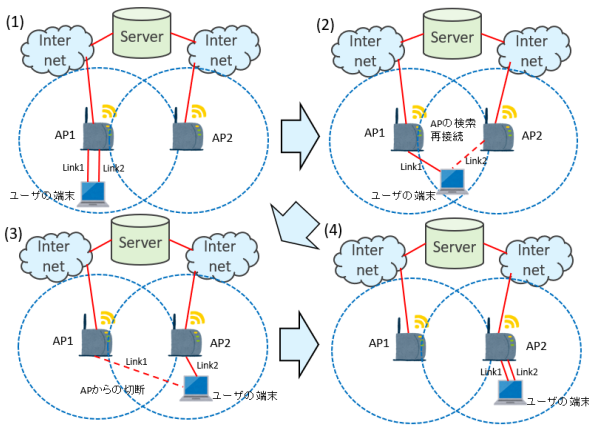


図 3 提案手法の概要

させている。図 1 に MPTCP の基本構成図を示す。MPTCP 環境が実装されている端末同士で通信を行った際に、複数 IP アドレスを持っている端末が存在した場合、新たなサブフローの接続が確立される。MPTCP 接続の確立は TCP と同様に 3Way-Handshake によって行われるが、そのすべてのパケットに MP_CAPABLE オプションを含ませており、相手が MPTCP に対応しているかの確認を行う。2 つのインタフェースが通信を確立するまでの流れを図 2 に示す。まず、1 つ目のインタフェースがサーバ側に SYN パケットと共に MP_CAPABLE オプションを送信する。このとき、サーバ側が MPTCP に対応していない場合には通常の TCP 通信を行う。1 つ目のインタフェースでの接続が確立されると、2 つ目のインタフェースが持つ IP アドレスを事前にサーバ側に送信する。この際、サーバ側に ADD_ADDR オプションと 2 つ目のインタフェースの持つ IP アドレスを送信する。ADD_ADDR オプションを付与したパケットと共に 2 つ目のインタフェースの持つ IP アドレスを送信することで、1 つ目のインタフェースと 2 つ目のインタフェースが同じ端末からのアクセスであることが確認できる。その後、2 つ目のインタフェースで 3Way-Handshake を行い、接続を確立する。この際、全てのパケットに MP_JOIN のオプション

を付与して送受信する。2 つ目のインタフェースがサーバ側に SYN パケットを送る際に、Address ID を送信し、ADD_ADDR オプションで送ったパケットに付与していた ID と同じであるかを判別する。この ID が一致した場合、これらのインタフェースが同じ端末から送信されていると判断できる。この確認にはトークンを用い、ADD_ADDR の際に送られてくる。このようにして、2 つのサブフローでの接続が確立され、複数のインタフェースで通信を行うことができる。

4. 提案手法

本稿では、エンドツーエンドでのハンドオーバを、MPTCP を用いて実現する手法を提案する。提案手法を図 3 に示す。まず、(1)のように、AP1 に 2 本の TCP 接続を確立する。その後 AP1 の範囲から出るように移動し、(2)のように 2 つのアクセスポイントの電波が重なる範囲に移動する。このとき、電波強度の値がこちらの定めた閾値を下回ると、まず Link2 の接続を切断し、電波強度のより強いアクセスポイントの探索を行う。元のアクセスポイントよりも電波強度の強いアクセスポイントが見つかったら、再接続処理を行い、通信を再開する。その間、Link1 の通信は継続する。最初のアクセスポイントから完全に出た後、(3)のように Link1 の接続も繋ぎ直し、最終的には(4)に示すように AP2 に両方とも接続を繋ぎ、安定した通信を行う。

安定した通信を行いつつ正確にハンドオーバーするために、本手法では新しいアクセスポイントへの通信が確立してから数秒後に 2 つ目のインタフェースが再接続を開始する。今回は安定性を求めたため、ある一定より電波が弱くなると 1 つ目のインタフェースがハンドオーバーを行い、完了した後に数秒後にもう 1 つのインタフェースもハンドオーバーを行う。

5. 実験と性能評価

5.1 実験準備と条件

今回の実験は、Linux 上で行った。また、Wireshark を用い、スループットの値と RTT を測定した。使用した OS は

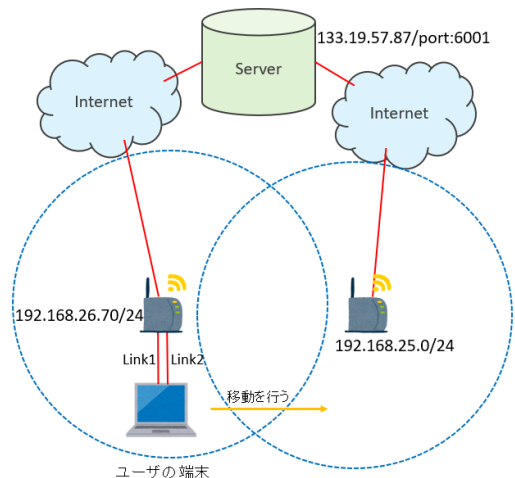


図 4 行った実験の構成図

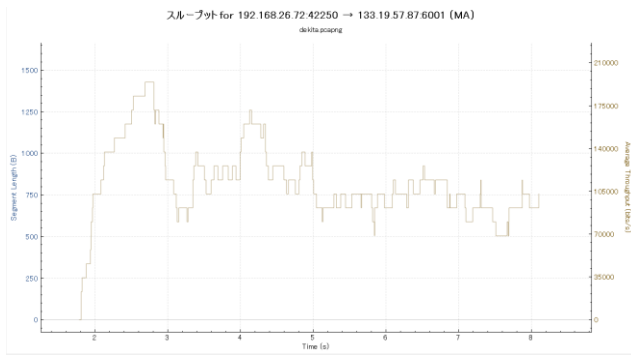


図 5 192.168.26.72 のスループット

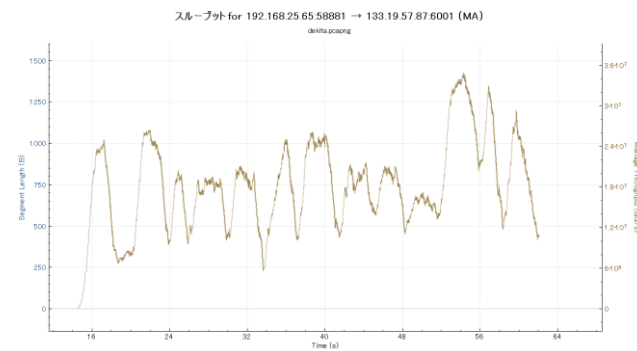


図 6 192.168.25.65 のスループット

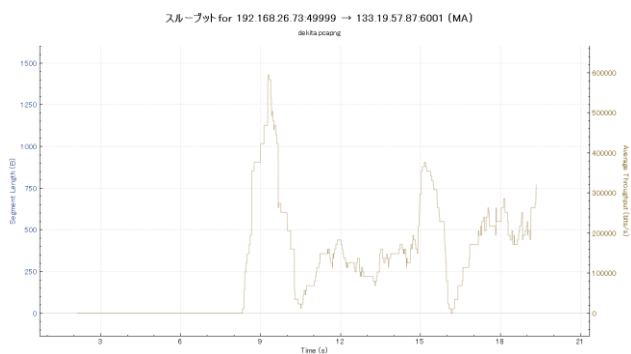


図 7 192.168.26.73 のスループット

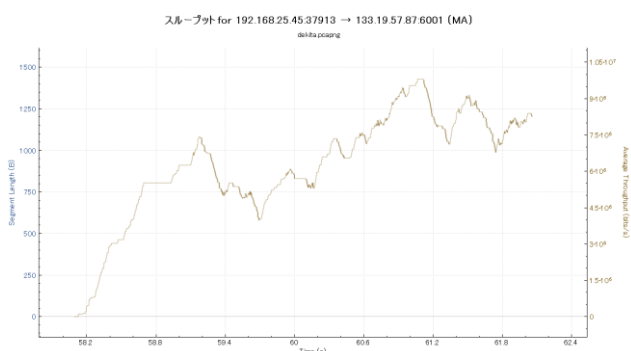


図 8 192.168.25.45 のスループット

Ubuntu16.04, kernel は 4.14.105.mptcp である。実験環境は図 4 に示すとおりであり、サーバとユーザ端末間で Iperf による通信を行いながらハンドオーバーを行い、Iperf が途中で停止しないことと Wireshark でハンドオーバー後も通信が

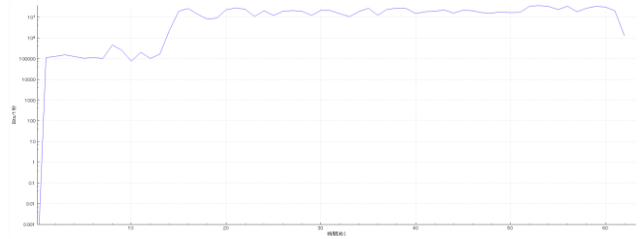


図 9 全体のスループット

途絶しないことをもって、期待する動作が行われたと判定するものとした。また、各アクセスポイントのセグメントは図 4 の通りである。

また、今回は、2 台のアクセスポイントを使用して、最初は同じアクセスポイントに 2 つとも接続し、その後片方ずつアクセスポイントを切り替えていく。具体的には、Link1 が 192.168.26.72 から 192.168.25.45 へのハンドオーバー、Link2 が 192.168.26.73 から 192.168.25.65 へのハンドオーバーである。なお、計測時間は 60 秒とした。

5.2 実験結果

以下に実験結果を示す。今回の通信では、図 5, 6, に示す、192.168.26.72 から 192.168.25.65 へのハンドオーバーと、図 7, 8 に示す、192.168.26.73 から 192.168.25.45 へのハンドオーバーを行った。通信が始まると、まず 192.168.26.72 とサーバ側での接続が確立された。そのため、こちら側をメインとして通信を行ったことが分かる。また、これらのスループットの値の確認をすると、一度も通信が切れないことが分かった。また、全体のスループットを確認すると、図 9 のようになった。これを確認すると、Iperf を開始してから終了までの間の 60 秒間において、一切通信が途切れていないことが分かる。

また、IP アドレスごとの接続の RTT を測定すると、以下の図 10.11.12.13 のようになった。セグメントが 25 である IP アドレスのときの方が、セグメントが 26 である IP アドレスのときより RTT の値が小さいことが分かった。

5.3 考察

今回の実験により、MPTCP を用いたハンドオーバーを行うことで、通信が途切れることなくアクセスポイントを切り替えることができた。しかし、計測結果を見ると、1 つ目のインタフェースの繋ぎ直しの際に、開始から 8s 後に切れて 16s 後に再接続、つまり約 8s かかったのに対して、2 つ目のインタフェースの繋ぎ直しの際は開始 20s 後に切れて 58s 後に再接続、つまり約 38s かかっていることが分かる。2 つ目のインタフェースの繋ぎ直しなので、もう一方が新しいアクセスポイントに接続して通信は行っているが、現在使用しているアクセスポイントからの Wi-Fi が届くのが半径約 15m であり、38s もかかるとアクセスポイントの範囲を横切るのに十分な時間がかかってしまう。すると、1 つ目のインタフェースがさらに新しいアクセスポイントにハンドオーバーを行おうとすると、通信が途切れてしまう。そのため、どちらのインタフェースの繋ぎ直しも 10s 以内に行う必要があると考える。

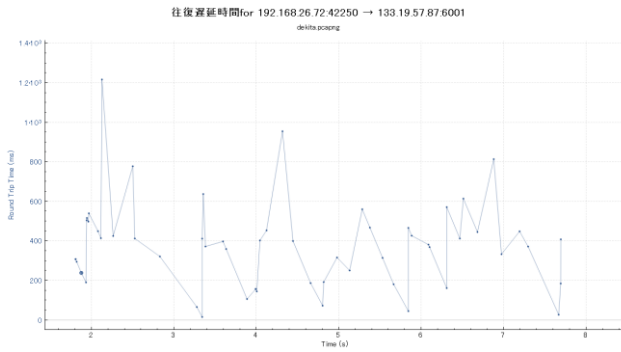


図 10 192.168.26.72 の RTT

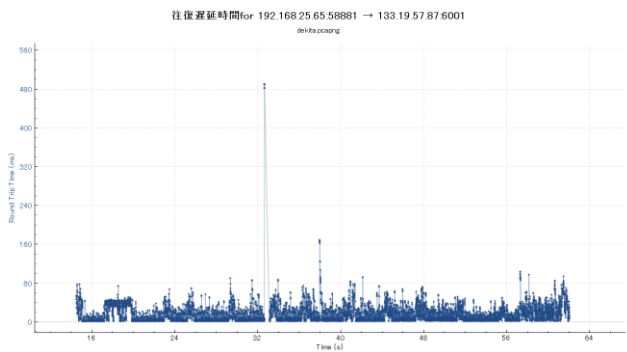


図 11 192.168.25.65 の RTT

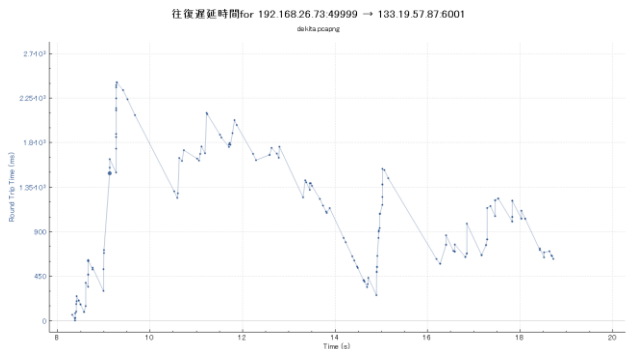


図 12 192.168.26.72 の RTT

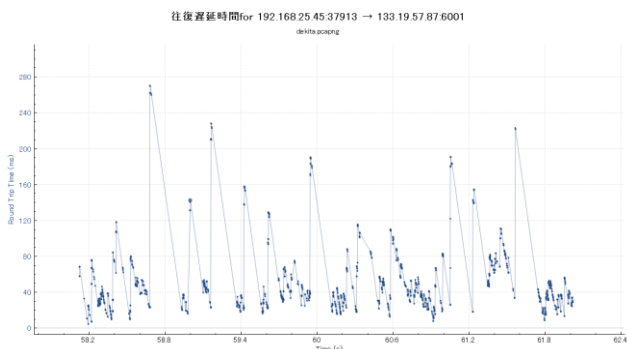


図 13 192.168.25.45 の RTT

また、RTT の計測結果より、メインだと思われるコネクションの方が全体的に RTT の値が小さかったため、複数の TCP コネクションを使用している場合でもメインで使用されるサブフローがあることが分かった。

6. おわりに

本稿では、TCP の拡張である MPTCP を用いてハンドオーバーを行う手法を提案した。複数のインターフェースで通信をすることにより、アクセスポイントが切り替わる際に片方だけの切り替えを行うと、もう一方での通信は継続されるため、途切れることなく通信を継続することができるというものである。

実際に動作を確認すると、繋ぎ直しに時間はかかるものの、スループットの値を確認すると通信が継続していることが確認できた。また、セグメントごとに RTT の値が違ってくることも測定できた。今回は通信の継続性のみの確認となってしまうため、これからはスループットの安定と共に RTT の値も安定させるように性能向上させるための手法の改善も行いたいと考えている。

謝辞

本研究は JSPS 科研費 JP18K18045 の助成を受けたものです。

参考文献

- [1] Alan Ford, Costin Raiciu, Mark J. Handley, and Olivier Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses", rfc6824, <https://www.rfc-editor.org/rfc/rfc6824.txt> (2013).
- [2] C.Perkins, Ed., "IP Mobility Support for IPv4", rfc3344, <https://www.rfc-editor.org/rfc/rfc3344.txt> (2002).
- [3] 金本綾子, 鈴木秀和, 伊藤将志, 渡邊晃, "IPv4 移動体通信システムにおけるパケットロスレスハンドオーバーの提案", 情報処理学会論文誌, Vol.50, No.1, pp.133-143, (2009).
- [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, No.12, pp.3243-3257, (2006).