

マルチホーム環境における MPTCP による複数端末間トラフィック分散機能の性能評価 Performance Evaluation of Traffic Distribution System among Multiple Terminals Using MPTCP in Multi-homed Network Environment

浅倉 龍次[†] 堀込 怜土[†] 山井 成良[†] 北川 直哉[†] 大坐島 智[‡]
Ryuji Asakura Reido Horigome Nariyoshi Yamai Naoya Kitagawa Satoshi Ohzahata

1. はじめに

無線通信の規格として Wi-Fi が広く利用されている。Wi-Fi では、各無線端末がアクセスポイント(AP)に接続して通信を行う。各無線端末は協調せずに AP との接続を行うため、AP に周囲に利用可能な AP が複数ある状況でも一部の AP に無線端末が集中して通信を行う場合がある。これは各 AP のトラフィックに偏りを発生させ、一部の AP に大きな負荷をかける可能性がある。Wi-Fi では AP のトラフィック分散を行うことでスループットが改善されることが報告されており [1], AP のトラフィック分散は高速な無線通信を実現するうえで重要であるといえる。

本研究では、Multipath TCP (MPTCP) [2]を使用して無線端末の接続先 AP を変更することによるトラフィック分散を考える。MPTCP は、複数経路を用いた通信を行うマルチホーミング通信に対応した TCP 拡張の protocols である。MPTCP では経路ごとにサブフローと呼ばれる TCP コネクションを持ち、それらの集合が一つの MPTCP コネクションとなる。サブフローにはアクティブ、バックアップの二つの優先度を設定することができ、通常はアクティブに設定されたサブフローを使用し、アクティブの経路が使用不可の場合のみバックアップに設定されたサブフローを使用する、といった通信が可能になる。本研究では、MPTCP の優先度変更によって通信を行うサブフローを変更できることに着目し、サブフローの切り替えによって無線端末が使用する AP を切り替えることによるトラフィック分散機能を提案する。

以降、第 2 章で提案手法について説明し、第 3 章で提案手法の性能評価と考察を行い、第 4 章で結論と今後の課題を述べる。

2. 複数端末間トラフィック分散機能

2.1 ネットワーク構成

本研究チームでは、無線区間に MPTCP を適用するために図 1 のようなネットワーク構成を提案している [3] [4]。VPN サーバと無線端末を MPTCP で接続することによって、無線区間でマルチホーミング通信を行えるようにしている。また、この構成によって VPN サーバが AP の混雑状況を監視しながら経路を制御できるようになる。

2.2 経路切り替え

経路の切り替えは MPTCP の優先度設定の変更によって行う。まず、一つのサブフローをアクティブ、もう一つのサブフローをバックアップとして設定しておく。VPN サー

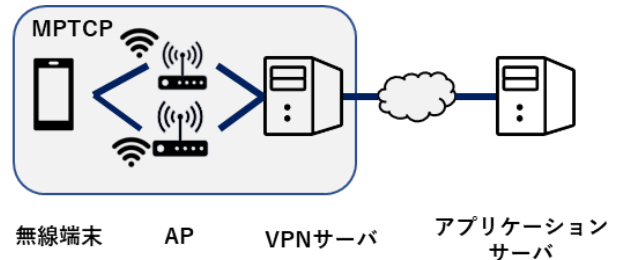


図 1 提案ネットワーク構成

バは特定の AP が混雑していると判断したときに、その AP を使用している無線端末一つを選択する。VPN サーバはその無線端末とのサブフローのうちアクティブに設定していたサブフローをバックアップに、バックアップに設定していたサブフローをアクティブに設定する。無線端末は VPN サーバからパケットを受信したときに MPTCP ヘッダーのサブフローの優先度変更を通知するオプションを参照することによってサブフローの優先度変更を認識できる。このようにして混雑している AP を使用している無線端末のアクティブなサブフローを別の AP を通るサブフローに変更することによって、経路を切り替えてトラフィックを分散する。

2.3 AP の混雑判断

AP の混雑状況の判断基準として、AP の無線端末からの受信バイト数を使用する。経路切り替えのアルゴリズムをアルゴリズム 1 に、各変数の意味を表 1 に示す。受信バイト数から AP の混雑状況が判断できるということは 3.1 で確かめる。

2.4 実装

本研究では複数端末間トラフィック分散機能をネットワークシミュレータ ns-3 [5]で実装した。標準の ns-3 では MPTCP は未実装であり MPTCP 通信をシミュレートできない。そこで、Linux カーネルと Linux 上で動作するアプリケーションを ns-3 上で利用可能にする ns-3 Direct Code Execution (DCE) [6]と、任意の Linux カーネルのネットワークスタックを共有ライブラリとして利用可能にする library operating system version of Linux kernel (LibOS) [7] [8]を ns-3 上に実装した。MPTCP カーネル [9]を LibOS に組み込み、LibOS を DCE から呼び出すことによって ns-3 上で MPTCP をシミュレーションできるようにした。

3. 実験

提案手法の性能評価をするために、本研究では ns-3 を用いたシミュレーションによって実験を行った。

[†] 東京農工大学 Tokyo University of Agriculture and Technology

[‡] 電気通信大学 The University of Electro-Communications

アルゴリズム 1 トラフィック分散アルゴリズム

```

 $R_{\alpha}(t) = Rx_1(t)/Rx_2(t), R_{\beta}(t) = Rx_2(t)/Rx_1(t)$ 
if  $R_{\alpha}(t) > T$  then
    (AP1にトラフィックが集中と判断)
    「AP1と接続している」かつ「AP2のRSSI>S」の
    端末を一つ選択
    選択した端末のAP1のサブフローをバックアップ,
    AP2のサブフローをアクティブに設定
else if  $R_{\beta}(t) > T$  then
    (AP2にトラフィックが集中と判断)
    「AP2と接続している」かつ「AP1のRSSI>S」の
    端末を一つ選択
    選択した端末のAP2のサブフローをバックアップ,
    AP1のサブフローをアクティブに設定
Endif
    
```

表 1 アルゴリズム 1 の変数

変数名	意味
$Rx_i(t)$	AP _i の受信バイト数
T	混雑していると判断する閾値
S	通信可能だと判断する RSSI の閾値

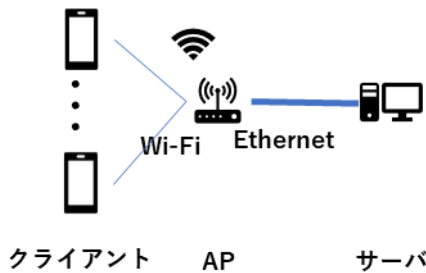


図 2 実験 3.1 のネットワーク構成

3.1 AP 混雑の判断

提案アルゴリズムでは、混雑の判断基準として AP の無線端末からの受信バイト数を使用している。この実験では、受信バイト数から AP の混雑状況を推察できることを確認する。実験を行うネットワーク構成を図 2 に示す。各クライアントは AP を介してサーバと接続し、サーバがそれぞれのクライアントへデータを送信している。この状態のとき、クライアント数が増えると AP の受信バイト数がどのように変化するかを計測する。

実験結果を図 3 に示す。端末数の増加に伴い受信バイト数が増加していることがわかる。よって、受信バイト数の増加から AP の混雑状況を推定できるといえる。

3.2 無線端末-AP 間距離と RSSI

経路切り替えを行う際、無線端末と切り替え先の AP の通信状況が悪いと通信が不可能になってしまう可能性がある。この問題を回避するために、アルゴリズム 1 では切り替え先の AP が無線端末から受信したときの RSSI を見て、一定値以下の場合はその無線端末では切り替えが行われないうようにしている。この実験では、通信状況が悪いと判断するための具体的な RSSI を検証する。実験のネットワー

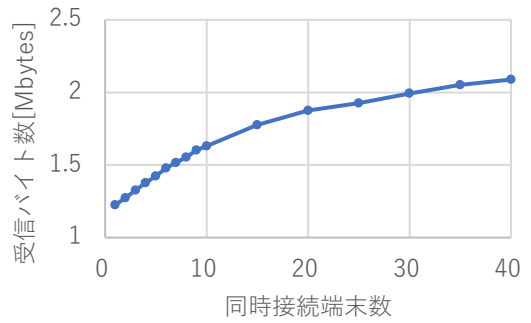


図 3 実験 3.1 の結果

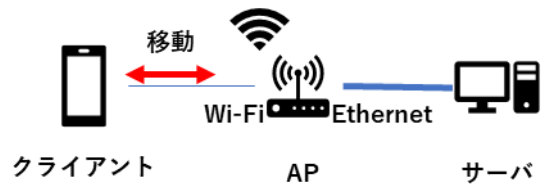


図 4 実験 3.2 のネットワーク構成

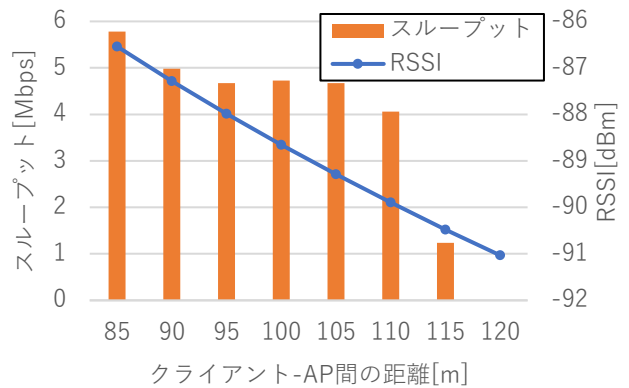


図 5 実験 3.2 の結果

ク構成を図 4 に示す。AP を介して一台のクライアントをサーバと接続し、サーバからクライアントへデータを送信する。このとき、クライアントと AP の距離を変化させ、スループットと RSSI がどのように変化するかを計測する。

実験結果を図 5 に示す。クライアント-AP 間の距離が 110m 付近まではスループットと RSSI 共に緩やかに低下しているが、115m からスループットが著しく低下していることがわかる。110m のときの RSSI は -90dBm で、少なくとも RSSI がこれ以上大きい場合ならば良好な通信が可能であると考えられるため、-90dBm を経路切り替えが可能閾値とする。

3.3 提案手法の性能評価

実験のネットワーク構成を図 6 に示す。各クライアントは AP を介してサーバと接続し、サーバからクライアントへデータを送信する。また、AP を二つ設置し片方は一台のクライアント、もう片方は複数のクライアントを接続する。このようにして片方の AP にトラフィックが集中している状況を再現する。このときに提案手法を適用していない場合と提案手法を適用したときの各クライアントのスループットを測定し、提案手法によって AP のトラフィック

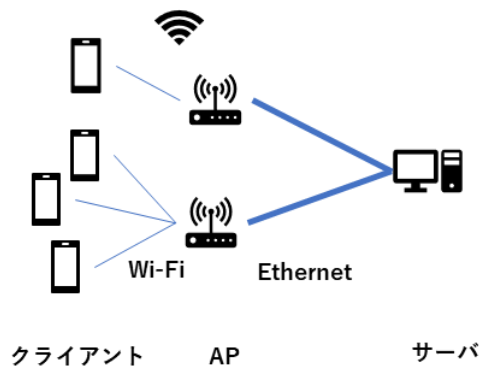


図6 実験3.3のネットワーク構成

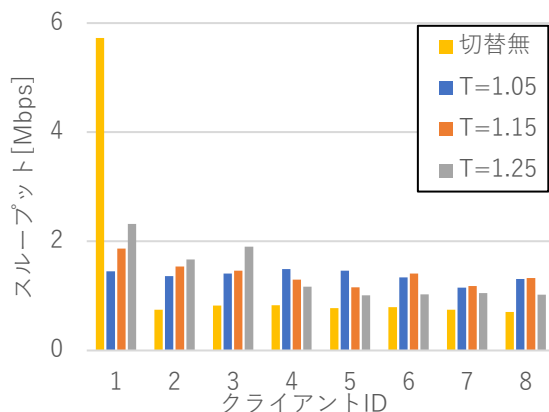


図7 実験3.3の結果（クライアント8台）

集中が解消されるかを確認する。アルゴリズム1で示したトラフィックが集中していると判断する変数 T の値が 1.15, 1.25, 1.35 の三パターンの場合で検証を行う。

クライアントが8台の場合のスループットを図7に示す。従来の通信方式では片方のAPにトラフィックが集中したままであるため、クライアント1のみ約5.7Mbpsのスループットが出ているがそのほかのクライアントは0.8Mbps程度のスループットしか出ていない。提案手法を使った場合は、 T がどの値のときでもすべてのクライアントのスループットが1Mbpsを超えていることがわかる。特に $T=1.05$ のときは最もスループットが大きい端末で1.5Mbps、最もスループットが小さいクライアントで1.2Mbpsとなっておりクライアント間のスループットの差がほとんどなくなっていることが確認できる。 T ごとのサブフローの切り替え回数を表2に示す。閾値が小さいほどサブフローの切り替え回数が増えていることがわかる。 T が小さいときは受信バイト数に大きな差がない場合でもサブフロー切り替えが行われるため、トラフィックの偏りに敏感に反応して切り替えを行うことができる。しかし、トラフィックの偏りがほとんどない場合に不要なサブフロー切り替えが行われてしまう可能性がある。

クライアントが4台の場合のスループットを図8に、サブフローの切り替え回数を表2に示す。基本的な傾向はクライアントが8台のときと同じだが、 $T=1.25$ としたときサブフローの切り替えが一切行われず、トラフィック分散が行われていない。 T の値は提案手法の動作に大きな影響を与えており、適切な値を設定する必要があるといえる。

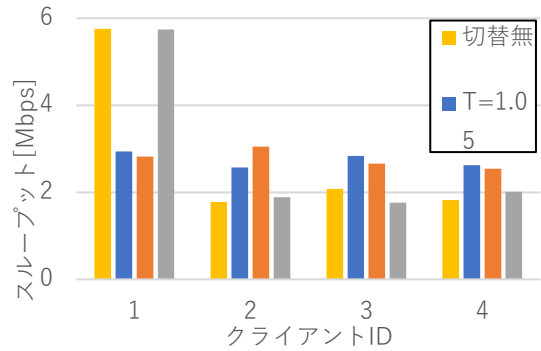


図8 実験3.3の結果（クライアント4台）

表2 サブフローの切り替え回数

	$T=1.05$	$T=1.15$	$T=1.25$
8台	40	20	2
4台	43	37	0

4. おわりに

Wi-Fi では、複数の利用可能なAPがある場合でも一部のAPにトラフィックが集中する可能性があるという問題がある。この問題を解決するために、本稿ではMPTCPを用いた複数端末間トラフィック分散機能を提案し、その性能評価を行った。その結果、提案手法によって各APのトラフィックが分散されることを確認できた。

今後の課題は、APが3台以上ある場合のトラフィック分散アルゴリズムの提案と、様々な環境での性能評価である。

謝辞

本研究の成果の一部は科学研究費助成事業：基盤研究(C) (課題番号17K00118) の助成による。

参考文献

- [1] L. Yen, T. Yeh and K. Chi, "Load Balancing in IEEE 802.11 Networks," *IEEE Internet Computing*, vol. 13, no. 1, pp. 56-64, Jan.-Feb. 2009.
- [2] A. Ford, C. Raiciu, M. J. Handley and O. Bonaventure, "TCP Extensions for Multipath Operation with Multiple Addresses," IETF RFC 6824, 2015.
- [3] Reido Horigome, Nariyoshi Yamai, Naoya Kitagawa, Satoshi Ohzahata, "Alternating Primary Subflow in MPTCP by External Program without Kernel Modification," in *Proceedings of 2018 IEEE 42nd International Conference on Computer Software and Applications (COMPSAC 2018)*, Tokyo, Japan, 2018.
- [4] 佐藤将斗, 堀込裕士, 浅倉龍次, 山井成良, 北川直哉, 大坐阜智, "任意のプロセスのMPTCPサブフロー情報取得および優先度設定を行うシステムコール群の実装," *IPJSJ 研究報告インターネットと運用技術 (IOT)*, Vols. 2019-IOT-44, no. 39, pp. 1 - 8, 2019.
- [5] "ns-3," [Online]. Available: <https://www.nsnam.org>.
- [6] H. Tazaki, F. Uarbani, E. Mancini, M. Lacage, D. Camara, T. Turetletti and W. Dabbous, "Direct code execution: Revisiting library os architecture for reproducible network experiments," in *Proceedings of the Ninth ACM Conference on Emerging Networking Experiments and Technologies, CoNEXT '13*, New York, NY, USA, 2013.
- [7] Yuji Sekiya Hajime Tazaki, Ryo Nakamura, "Library operating system with mainline linux network stack," in *netdev 0.1*, Ottawa, On, Canada, 2015.
- [8] "Network Stack in Userspace," [Online]. Available: <https://github.com/libos-nuse/net-next-nuse>.
- [9] "Multipath tcp linux kernel implementation," [Online]. Available: <http://multipath-tcp.org/>.