

# MPTCP を用いたマルチホーム環境における端末での常時バイキャストによる性能向上

松浦 穰<sup>†1,a)</sup> 山井 成良<sup>†1,b)</sup> 中川 令<sup>†1,c)</sup> 大坐 畠 智<sup>†2,d)</sup>

**概要:** 現在, モバイル通信トラフィックの約 63% を占めるほど Wi-Fi は一般的に利用されているが, このときに 2.4GHz 帯の電波干渉問題や隠れ端末問題が発生すると, パケットロスによってスループットが低下する. この問題を解決するために本論文ではマルチホーム通信に着目し, クライアント側からは MPTCP 常時バイキャスト通信を行い, 通信量の大きいサーバ側からは通常の MPTCP 通信を行う手法を提案した. その際, Open vSwitch のグループテーブルを用いてクライアント側からのパケットを複製することでバイキャスト通信を実装した. 性能評価のためにサイズの大きいファイルを転送する実験を行ったところ, 通常の MPTCP 通信よりも提案した MPTCP 常時バイキャスト通信の方が高速にファイル転送できることを確認した. また, クライアントからのパケットロス率が高いほど通常の MPTCP 通信と提案した常時バイキャスト通信のファイル転送にかかる時間差は大きくなったため, 無線アクセスポイント付近に電波を発生させる機器が存在するといった過酷な通信環境において, 本論文で提案したクライアントからの MPTCP 常時バイキャスト通信が有用になると考察した.

## Performance Improvement by Constant Bicast on Terminals in Multi-home Environment using MPTCP

### 1. はじめに

近年ではスマートフォンやタブレットなどの無線通信を利用するモバイル端末の普及が進んでおり, 総務省が毎年実施している通信利用動向調査によると, モバイル端末の世帯保有率が9割を超えている [1]. また, 現在広く利用されている無線通信規格として Wi-Fi が挙げられ, モバイル通信トラフィックの約 63% を占めている [2].

Wi-Fi はアクセスポイント (AP) を介して通信を行うインフラストラクチャモードでの利用が一般的であるが, このときに問題となるのが電波干渉である. これは 2.4GHz 帯の Wi-Fi で通信を行っているときに, 同じく 2.4GHz 帯の電波を利用する電子レンジを使用した場合, 電波干渉を

起こし通信速度を低下させる. このとき, 機種によっては AP と電子レンジの距離が 70m 以上離れていても通信速度が半減する.

また, 複数の無線端末がその中間に位置する AP に同時にデータを送信することで電波が衝突し, AP が正常にデータを受信できなくなる隠れ端末問題があり, これは学校など無線端末同士が互いの通信可能距離外に存在する場合に頻発する. RTS/CTS フレームを付けてパケットを送信することでこの問題を回避することは可能であるが, その場合は通信速度が大きく低下してしまう [3].

本論文ではこれらの問題を解決するためにマルチホーム通信に着目し, クライアント側からは MPTCP 常時バイキャスト通信を行い, 通信量の大きいサーバ側からは通常の MPTCP 通信を行う手法を提案する. マルチホーム通信とは, 複数の経路を同時に利用して通信を行うことで通信の効率や安定性を向上させる通信方式であり, それを実現するプロトコルとして TCP の拡張プロトコルである Multipath TCP (MPTCP) を採用した. バイキャスト通信については, Open vSwitch を用いてクライアントから送信されるパケットを複製することで常時バイキャスト

<sup>†1</sup> 現在, 東京農工大学  
Presently with Tokyo University of Agriculture and Technology

<sup>†2</sup> 現在, 電気通信大学  
Presently with The University of Electro-Communications

a) joh\_matsuura@net.cs.tuat.ac.jp

b) nyamai@cc.tuat.ac.jp

c) fr3660@go.tuat.ac.jp

d) ohzahata@is.uec.ac.jp

通信を実現する。Open vSwitch にはグループテーブルという機能が備わっており、これは入力されたパケットに対して複数のアクションを実行できるため、容易にパケットを複製して複数のサブフローに流すことができる。

## 2. 関連研究

### 2.1 Multipath TCP ( MPTCP )

MPTCP とは複数の TCP コネクションをサブフローとしてまとめて管理することで、マルチホームネットワークを実現する TCP の拡張プロトコルである。そのため TCP との高い互換性を持ち、TCP を利用しているアプリケーションに対して、MPTCP を利用するための変更は不要である。また、Linux カーネルに MPTCP を適用することが可能なため、本研究では Linux 環境における MPTCP マルチホーミング通信を用いて研究を行う。

MPTCP は通信の際にパケットを各サブフローに割り当てるスケジューラという機能を持つ。現在の MPTCP スケジューラには default , round-robin , redundant の 3 種類が存在するが、本研究では default スケジューラを利用する。これは、パケットの送受信にかかった時間を表す RTT (Round Trip Time) が最も小さいサブフロー 1 つを用いて通信を行う。そのサブフローの輻輳ウィンドウサイズを超過した場合は、異なるサブフローを用いた通信に切り替えられる。

### 2.2 Open vSwitch ( OVS )

OVS とは、オープンソースの Apache License 2.0 の下でライセンスされている仮想マルチレイヤソフトウェアスイッチである。OVS は OpenFlow の各種プロトコルに対応しており、OpenFlow コントローラと OVS を接続することで OpenFlow スイッチとして動作させることができるが、コントローラを用いずに手でフローエントリを OVS に設定することも可能である。本論文では動的にルーティングを変更する必要がないため、OpenFlow を用いずに OVS に直接フローエントリを設定することで常時バイキャスト機能を実現する。

#### 2.2.1 Group Table

グループテーブルとは、フローテーブルエントリでは実装が難しい、複雑なパケット操作を容易にする OVS の抽象化機能である [4]。各グループは入力としてパケットを受け取り、これらのパケットに対して任意の OpenFlow アクションを実行する。この機能を用いることで、ロードバランシングやマルチキャストなどに動作を実装することができる。

グループはパケットリストと呼ばれるアクションリストのリストを内包している。各パケットまたはパケットリストは、入力パケットに適用することができる。詳細な処理はグループの Type (種類) に依存し、Type によってはパ

ケット内の追加パラメータを利用する。Type には ALL, SELECT, INDIRECT, FAST-FAILOVER の 4 種類が存在するが、本論文では最も単純な ALL グループを利用する。これはパケットリスト内の各パケットで独立して操作できるように受信パケットを複製する。つまり、各パケットに異なるアクションを設定することで複数のコピーされたパケットに対してそれぞれ異なる操作を行うことができる。すべてのパケットに設定されたアクションが実行されるので、パラメータを設定する必要はない。

### 2.3 Wi-Fi

Wi-Fi は現在広く利用されている無線通信方式の一つで、モバイル通信トラフィックの約 63% を占めている [2]。Wi-Fi は物理層とデータリンク層について規定している IEEE802.11 規格に準拠して通信を行う。無線ネットワークにおいて、複数の無線端末が同時にデータを送信するとデータが衝突するため、Wi-Fi では Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) というメディアアクセス制御技術が用いられている。

#### 2.3.1 2.4GHz 帯の電波干渉

Wi-Fi の通信規格には 2.4GHz と 5GHz の 2 つの周波数帯が存在する。2.4GHz 帯は周波数が低いため、電波が遠方まで届きやすく、壁や床などの障害物にも強いという利点がある。しかし、2.4GHz 帯は Wi-Fi だけでなく、電子レンジなどの家電や IH クッキングヒーター、Bluetooth など、他の電子機器も利用する周波数帯である。そのため、2.4GHz の Wi-Fi アクセスポイント周辺に電子レンジが存在する場合、互いの電波が干渉し合うことで通信速度が低下する [5]。

#### 2.3.2 隠れ端末問題

Wi-Fi では CSMA/CA でメディアアクセス制御を行っているが、Wi-Fi ネットワーク全体には電波が届かない場合があり、ある無線端末がアクセスポイントにデータを送信しようとしたときに別の無線端末が送信中であることに気付かず、フレームの衝突が発生してしまう隠れ端末問題がある。隠れ端末問題によるパケット衝突を軽減するために、RTS/CTS という方式が IEEE802.11 において規定されている。RTS/CTS とは、まず無線端末 A がデータを送信する前に送信要求 (RTS) フレームに送信したいデータサイズの情報を格納して送信する。RTS フレームを受信したアクセスポイントは、送信確認 (CTS) フレームというアクセスポイントが送信するデータサイズの情報が格納されたフレームを Wi-Fi ネットワークにブロードキャストする。CTS フレームを受信した無線端末は、他の無線端末が送信を開始することがわかるため、そのデータサイズに相当する時間はデータの送信をしないことで隠れ端末問題を軽減している [6]。

しかし、RTS/CTS を用いた Wi-Fi 通信では、RTS フ

フレームと CTS フレームを送信する分、通常の Wi-Fi 通信よりもスループットが低下する。そのため、一般的に利用されている製品では、アクセスポイントが自分の送信したデータに対する ACK が複数回戻ってこない場合に RTS の送信を開始する。なお、Wi-Fi の RTS/CTS は Bluetooth や電子レンジなどの Wi-Fi で通信していない機器から発せられる 2.4GHz 帯の電波干渉による通信速度低下の問題に対して、無線端末は RTS フレームを送信し始めることでさらなる速度低下が引き起こされる [3]。

### 3. 提案手法

#### 3.1 概要

前述の無線 LAN 環境の 2 つの問題を解決するため、MPTCP 環境においてクライアントが常時バイキャスト通信を行う手法を提案する。ネットワークの概略図を図 1 に示す。クライアントと Web サーバの間に MPTCP を適用した VPN サーバを設置することで、VPN サーバは MPTCP 通信と TCP 通信の中継を行うため MPTCP を導入していない Web サーバともクライアントは MPTCP 通信を行うことができる。さらに設置した VPN サーバに先行研究のサブフロー切り替え機能を搭載している [7][8]。

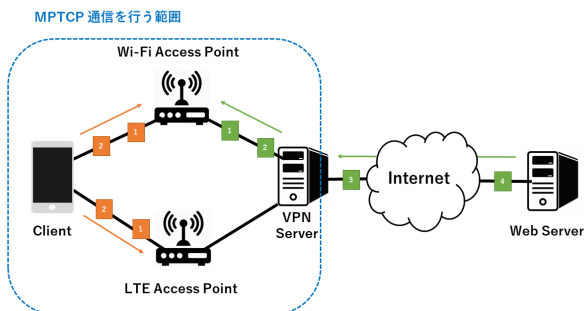


図 1 クライアント側のみ常時バイキャストを行う MPTCP 環境

本研究では、クライアントの常時バイキャストを実現するために、クライアントからパケットを送信する際に複製して送信元と宛先 MAC アドレスを変更したパケットを複数のサブフローに送信する手法を提案する。その際、クライアントとサーバの MPTCP スケジューラは default に設定するため、クライアント・サーバ間の MPTCP 通信で使用されるサブフローは 1 つであるが、ペイロードと送信元・宛先 IP アドレスが同一のパケットがクライアントから複数のサブフローに流れることになる。一方、サーバは複製されたパケットの送信元・宛先 IP アドレスが変更されていないため、実際にどのサブフローを通過しようとする問題なく通信を行うことができると考えた。

また、パケットを複製する処理には OVS を利用する。OVS は前述のグループテーブルという機能を備えており、これによって入力されたパケットに対して複数エントリの

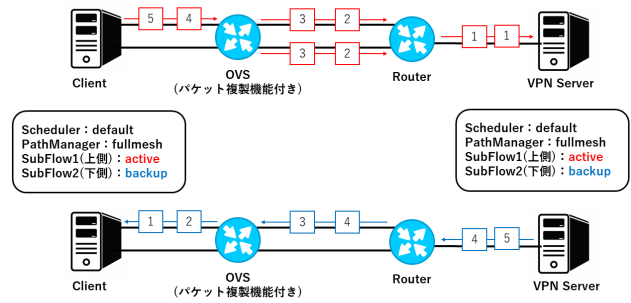


図 2 提案手法のパケットの流れ

実行が可能となるため、パケットの複製を容易に行うことができる。クライアントとサーバからのパケットの流れを図 2 に示す。なお、図 2 はサブフロー 1 を active に設定しているためサブフロー 1 にパケットが流れているが、反対にサブフロー 2 を active にした場合はサブフロー 2 にパケットが流れる。

#### 3.2 提案手法の実装

図 2 に示した、端末、サーバ、ルータ、OVS がそれぞれ 1 台ずつ存在する実験ネットワークを仮想マシン VirtualBox 上で実装する。このとき、クライアントとサーバの双方の MPTCP スケジューラを default に設定し、一方のサブフローを active に、もう一方のサブフローを backup に設定する。実装環境を表 1 に、各端末の IP アドレス、MAC アドレスを表 2 に示す。

表 1 実験ネットワークの実装環境

VirtualBox のバージョン		6.1.26
クライアント サーバ	OS	Ubuntu 16.04
	Linux バージョン	4.14.64
	MPTCP バージョン	0.95
ルータ	OS	VyOS 1.4.0
OVS	OS	Ubuntu 16.04
	Linux バージョン	4.14.64
	MPTCP バージョン	0.95
	Open vSwitch バージョン	2.5.9

表 2 各端末のアドレス情報

	IP アドレス	MAC アドレス
クライアント	192.168.110.3/24	08:00:27:86:0d:bc
	192.168.210.3/24	08:00:27:ec:c0:04
OVS	192.168.110.2/24	08:00:27:f9:49:62
	192.168.210.2/24	08:00:27:2a:a0:c3
	192.168.100.2/24	08:00:27:81:6b:cf
	192.168.200.2/24	08:00:27:e7:c6:f8
ルータ	192.168.100.1/24	08:00:27:33:6d:a7
	192.168.200.1/24	08:00:27:e1:9e:e3
	192.168.0.1/24	08:00:27:d9:74:68
サーバ	192.168.0.2/24	08:00:27:67:2b:63

### 3.3 提案手法の動作検証

クライアントとサーバ間で MPTCP コネクションを設立し、サーバが1秒間に1byteの packets を送信するプログラムを作成した。そのプログラムを実行中にサーバでサブフローの切り替えを行い、正常にクライアントの常時バイキャスト通信が実装できているかをパケットキャプチャで確認する。表3にサーバでパケットキャプチャした結果を示し、表4にクライアントでパケットキャプチャした結果を示す。

表3 サーバでのパケットキャプチャ結果

192.168.0.2 → 192.168.110.3: Flags [P.], seq 164:168, ack 42
192.168.110.3 → 192.168.0.2: Flags [P.], seq 42:43, ack 168
192.168.0.2 → 192.168.110.3: Flags [.] , ack 43
192.168.110.3 → 192.168.0.2: Flags [P.], seq 42:43, ack 168
192.168.0.2 → 192.168.110.3: Flags [.] , ack 43
～サブフローの切り替え～
192.168.0.2 → 192.168.210.3: Flags [P.], seq 24:28, ack 6
192.168.210.3 → 192.168.0.2: Flags [P.], seq 6:7, ack 28
192.168.0.2 → 192.168.210.3: Flags [.] , ack 7
192.168.210.3 → 192.168.0.2: Flags [P.], seq 6:7, ack 28
192.168.0.2 → 192.168.210.3: Flags [.] , ack 7

表4 クライアントでのパケットキャプチャ結果

192.168.0.2 → 192.168.110.3: Flags [P.], seq 164:168, ack 42
192.168.110.3 → 192.168.0.2: Flags [P.], seq 42:43, ack 168
192.168.0.2 → 192.168.110.3: Flags [.] , ack 43
192.168.0.2 → 192.168.110.3: Flags [.] , ack 43
～サブフローの切り替え～
192.168.0.2 → 192.168.210.3: Flags [P.], seq 24:28, ack 6
192.168.210.3 → 192.168.0.2: Flags [P.], seq 6:7, ack 28
192.168.0.2 → 192.168.210.3: Flags [.] , ack 7
192.168.0.2 → 192.168.210.3: Flags [.] , ack 7

表3と表4の赤と青で強調したIPアドレスは、それぞれクライアントのサブフロー1とサブフロー2に相当する。表3のマゼンタで強調した箇所に着目すると、サーバから送信された SYN パケットに対してクライアントが ACK パケットを返信するとき、OVSによって複製されたシーケンス番号が同一の2つの ACK パケットがサーバに到着していることがわかる。また、サーバがサブフローを切り替えても OVSによってクライアントの ACK パケットが複製されていることがわかる。表4のマゼンタで強調した箇所に着目すると、サーバから送信された SYN パケットに対してクライアントは ACK パケットを1つしか返信していないが、OVSによってパケットが複製されるため、サーバから2つの ACK パケットが送信されていることがわかる。

表3と表4のパケットキャプチャ結果から、OVSを用いたクライアントの常時バイキャスト通信が実装できていることが確認できた。

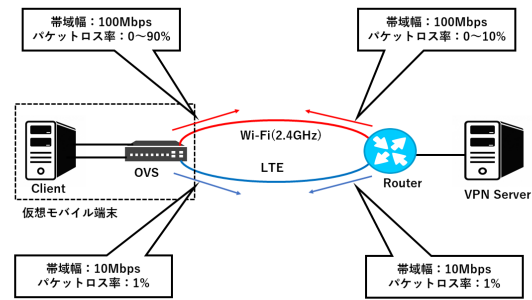


図3 OVS とルータの帯域幅とパケットロス率

## 4. 性能評価

### 4.1 実験の概要

本実験で想定している状況について述べる。まず、無線端末は Wi-Fi と LTE の電波範囲内に存在する。Wi-Fi を用いた無線通信は高速であるが、アクセスポイントの周辺に電波を発する機器が存在し、かつ無線端末同士が互いの認識範囲外に多数存在し隠れ端末問題が起りやすい状況、つまり外部からの電波干渉により通信が安定しない環境を想定する。また、LTE を用いた無線通信は安定した通信が可能であるが、利用者数が多いため低速に制限されている環境を想定する。

また、無線端末はバイキャスト通信機能を備えているため、前章で実装したクライアントと OVS を1つの無線端末とみなす。さらに、先に述べた Wi-Fi と LTE の無線通信状況をシミュレートするため、OVS とルータ間の2つの通信経路に帯域幅とパケットロス率を設定する。OVS 側は tc コマンドで設定し、ルータ側は VyOS の NetworkEmulator 機能で TrafficPolicy を設定する。図3に実験ネットワークと OVS ・ルータの帯域幅・パケットロス率について示す。

このネットワーク環境において、無線端末が MPTCP 通信を利用してサーバから大容量のファイルをダウンロードする時間を測定する。無線端末が常時バイキャストを行うことでどれだけファイル転送時間を短縮できるかを評価するため、通常の MPTCP 通信によるファイル転送時間も測定する。

### 4.2 実験の手順

実験の手順を以下に示す。

- (1) サーバで 500MB のデータファイルを作成し、apache を起動してダウンロード可能な状態にする。OVS はバイキャスト機能を ON に設定する。
- (2) ルータ側と OVS 側の両方で Wi-Fi のパケットロス率を 0% に、帯域幅は 100Mbps に設定する。LTE のロス率は双方で 1% に、帯域幅は 10Mbps に設定する。
- (3) クライアントはサーバから 500MB のデータファイル

をダウンロードし、そのファイル転送時間を測定する。  
これを複数回行い転送時間の平均値をとる。

- (4) OVS 側の Wi-Fi パケットロス率を 1% 上昇させ、手順 3 に戻る。ロス率が 10% の場合は手順 5 に進む。
- (5) ルータ側の Wi-Fi パケットロス率を 1% 上昇させ、OVS 側の Wi-Fi パケットロス率を 0% に設定し、手順 3 に戻る。ルータ 側のロス率が 5% の場合は手順 6 に進む。
- (6) OVS のバイキャスト機能を OFF に設定（通常の MPTCP 通信に設定）し、手順 2 へ戻る。既に OFF の場合は実験を終了する。

### 4.3 実験結果

クライアントがサーバから 500MB のデータファイルの転送にかかった時間について、提案手法の常時バイキャストの結果を表 5 に、通常の MPTCP 通信の結果を表 6 に示す。それぞれ縦方向が OVS からのパケットロス率 (単位: %), 横方向がルータからのパケットロス率 (単位: %) を表す。

表 5 常時バイキャスト通信でのファイル転送時間 (秒)

	0	1	2	3	4	5
0	41.4	41.0	41.2	44.2	50.2	61.4
1	41.4	41.0	41.6	44.6	50.2	61.0
2	41.6	41.0	41.6	44.6	50.2	60.0
3	41.8	41.0	41.8	44.6	50.0	62.6
4	42.0	41.2	41.8	44.6	50.0	60.0
5	42.0	41.2	41.7	45.0	49.8	62.0
6	42.2	40.8	41.7	45.0	51.0	59.3
7	42.2	41.0	41.7	45.3	49.7	59.3
8	42.2	40.8	41.3	45.0	52.0	61.7
9	42.2	41.0	41.7	45.7	52.3	62.7
10	42.2	40.8	42.0	45.7	53.7	61.3

表 6 通常の MPTCP 通信でのファイル転送時間 (秒)

	0	1	2	3	4	5
0	42.2	41.0	43.4	49.2	62.8	80.6
1	42.2	41.2	43.6	49.8	65.2	82.0
2	42.2	42.0	44.6	51.0	67.2	83.6
3	42.6	42.2	45.2	52.6	69.2	87.8
4	42.6	42.2	45.8	55.0	70.2	90.6
5	42.6	43.0	47.4	56.4	72.8	98.4
6	42.6	44.2	47.8	62.2	78.8	108.0
7	42.6	45.4	50.8	64.8	84.4	113.6
8	42.6	45.8	53.8	69.2	89.4	127.8
9	43.0	47.4	59.8	76.2	97.4	130.6
10	43.4	49.2	64.0	84.4	108.0	141.4

表 5 と表 6 をルータからのパケットロス率ごとグラフ化し、提案手法の常時バイキャスト通信と通常の MPTCP 通信を比較した図を図 4 から図 9 に示す。

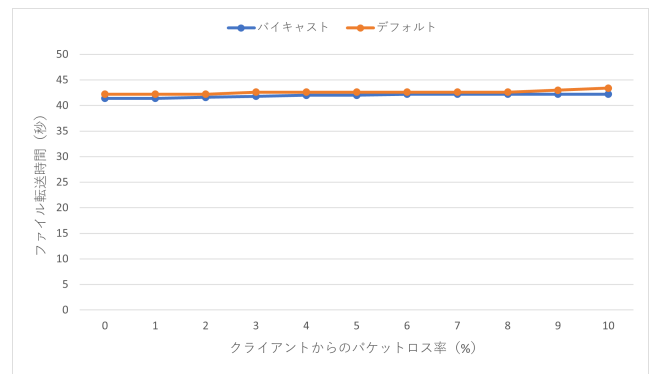


図 4 ルータからのパケットロス率 0% における比較

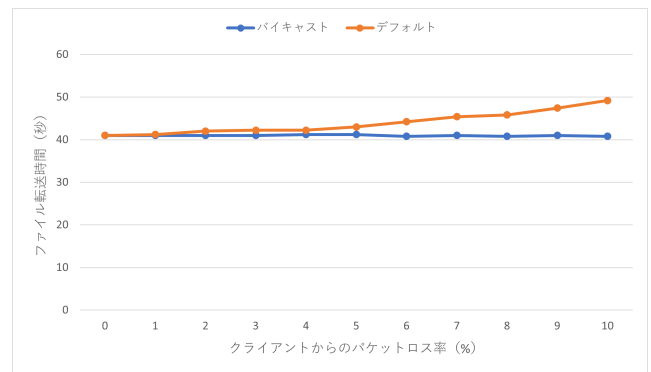


図 5 ルータからのパケットロス率 1% における比較

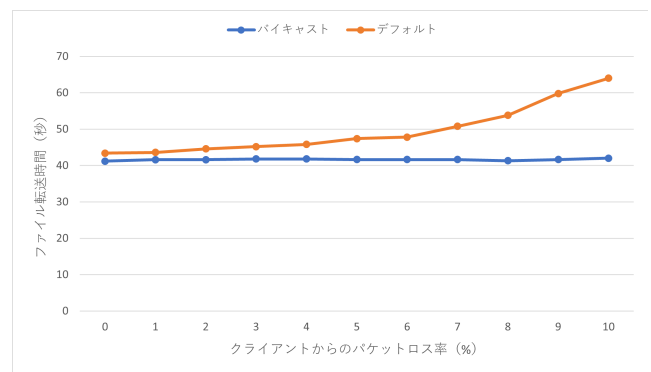


図 6 ルータからのパケットロス率 2% における比較

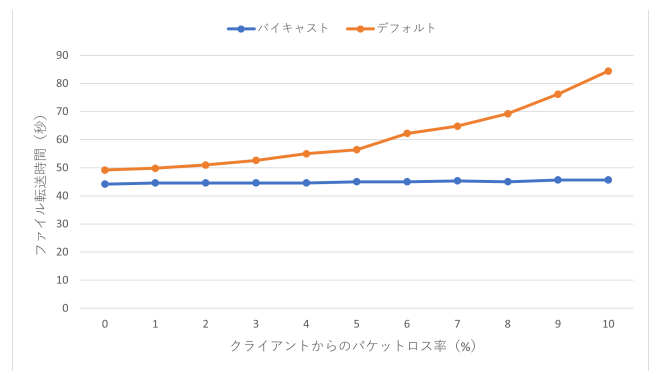


図 7 ルータからのパケットロス率 3% における比較

### 4.4 考察

4.3 の実験結果から、ルータからのパケットロス率に関

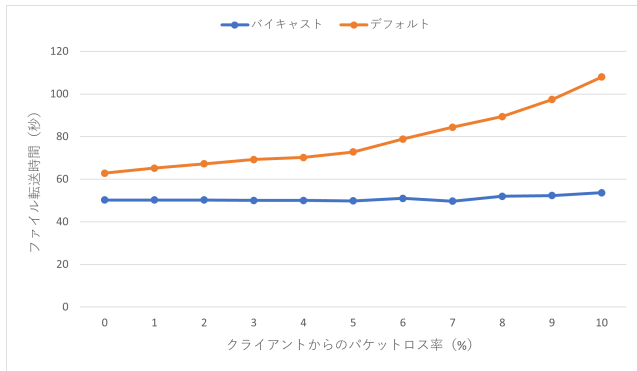


図 8 ルータからのパケットロス率 4%における比較

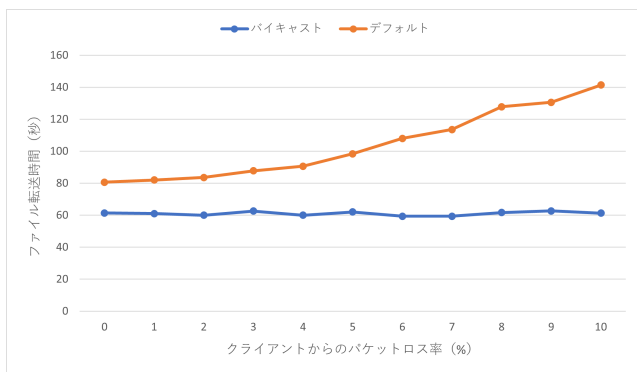


図 9 ルータからのパケットロス率 5%における比較

係なく提案手法であるクライアントの常時バイキャスト通信の方が通常の MPTCP 通信よりも高速にサーバからファイルを転送できることがわかる。また、クライアントまたはサーバからのパケットロス率が高くなるほど通常の MPTCP 通信は通信速度が低下しているのに対し、提案手法は 1 秒程度の低下に留まっていることがわかる。このことから、隠れ端末問題や外部の電子機器から発せられる電波による干渉によってクライアントからのパケットロス率が高い不安定な Wi-Fi ネットワーク環境において、提案手法であるクライアントの常時バイキャスト通信は通常の MPTCP 通信よりも高速なファイル転送が可能であるため、有用であると考えられる。

考察として、Wi-Fi ネットワークのクライアントからのパケットロス率が 0% の時、提案手法の常時バイキャスト通信と通常の MPTCP 通信でファイル転送の時間差が生じている点が挙げられる。この原因は、バイキャストによってクライアントからの ACK パケット数が増加したことにより、輻輳発生時に Fast Retransmission アルゴリズムが通常の MPTCP 通信時よりも高速にウィンドウサイズを増加させているからであると考えた。今後は ACK パケットとウィンドウサイズの関係性を明らかにするため、通信の解析を進めることを予定している。

## 5. おわりに

本論文では Wi-Fi ネットワークにおいて、隠れ端末問題

や Wi-Fi アクセスポイント周辺の電子機器による電波干渉などにより通信速度が低下する問題を解決するために、クライアントからの MPTCP 通信を常時バイキャストする手法を考案した。バイキャストを実現するため、OVS のグループテーブル機能を用いてクライアントから送信されるパケットを複製した。

提案手法の実装は仮想化ソフトウェアの VirtualBox 上で行い、通信経路のスループットとパケットロス率を設定することで Wi-Fi・LTE ネットワーク環境を再現し、サーバからファイルサイズの大きいデータを転送する時間について、常時バイキャスト通信と通常の MPTCP 通信でそれぞれ測定した。

その結果、提案手法であるクライアントの常時バイキャストの方が通常の MPTCP 通信よりも高速にファイル転送ができることを確認した。また、クライアントとサーバのパケットロス率が高くなるほどファイル転送の時間差は大きくなることが明らかになった。このことから、クライアントからのパケットロス率が高い不安定な Wi-Fi ネットワーク環境において、クライアントの常時バイキャストは通常の MPTCP 通信よりも高速な通信が可能となると評価した。

**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)(17K00118)「アクセスクラウド：複数の無線系ネットワークを活用した公平かつ高速な通信」の助成による。

## 参考文献

- [1] 総務省 | 令和 3 年版 情報通信白書 | デジタル活用の現状, 入手先 <https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/r03/html/nd111100.html> (2022/01/27).
- [2] Value of Wi-Fi — Wi-Fi Alliance, 入手先 <https://www.wi-fi.org/discover-wi-fi/value-of-wi-fi> (2022/01/27).
- [3] RTS/CTS と隠れ端末問題 — フィールドデザイン, 入手先 [https://fielddesign.jp/technology/wlan/ieee802\\_hidden/](https://fielddesign.jp/technology/wlan/ieee802_hidden/) (2022/01/27).
- [4] Creating a Group Table - PICOS 2.11.1 Configuration Guide - PICOS Documentation, 入手先 <https://docs.pica8.com/display/PICOS2111cg/Creating+a+Group+Table> (2022/01/27).
- [5] 守倉正博, 松江英明: *IEEE 802.11 準拠無線 LAN の動向*, 電子情報通信学会論文誌 B, Vol. J84-B, No. 11, pp. 1918-1927 (2001).
- [6] 藤原淳, 杉山隆利: 無線メッシュネットワークにおけるさらし端末問題を防ぐチャンネル割当法, *NTT DoCoMo テクニカル・ジャーナル*, Vol. 14, No. 4, pp. 53-56 (2007).
- [7] 堀込怜士, 浅倉龍次, 山井成良, 北川直哉, 大坐畑智: *MPTCP* における外部プログラムによるプライマリサブフローの切替, 研究報告インターネットと運用技術 (IOT), Vol. 40, pp. 1-6 (2018).
- [8] 佐藤将斗, 山井成良, 北川直哉, 大坐畑智: 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, 研究報告インターネットと運用技術 (IOT), Vol. 39, pp. 1-8 (2019).