

Regular TCP と MultiPath TCP のスループット比較

2012SE070 伊藤 寛晃

指導教員 後藤 邦夫

1 はじめに

近年、ネットワーク利用者は増大し、ネットワーク通信量も増大しており、通信の遅延や切断が問題となっている。だが、ネットワーク使用者の増加によりモバイル Wi-fi ルータや、公衆無線 LAN の設置、複数の ISP を用いるなど複数の経路を利用しやすい環境もできている。そこで複数の経路を用いることにより、ネットワークの遅延や切断が緩和されると考えた。

本研究では、複数の経路を同時に用いて通信すると、どの程度通信の遅延や切断が改善されるかをスループット比較する。複数の経路で行う通信プロトコルの代表的なものに SCTP [4] があり、研究されてきたが、これは中間ノード特に NAT で破棄される場合があり、普及しにくい。そこで本研究では、TCP のオプション 30(MPTCP) を利用して複数の経路で通信を行うため中間ノードで破棄されない MultiPath TCP [2] を用いて研究を行った。[1] トラフィック混雑時や接続障害時の遅延やパケット損失を想定し、エミュレータを用いて遅延やパケット損失を実装した上で通信実験を行い、それぞれのスループットを比較した。実験では Regular TCP を MultiPath TCP に変えた場合、どの程度性能が上がるのか実験、比較した。

2 MultiPath TCP

MultiPath TCP とは TCP オプション 30MPTCP を使用し、Regular TCP を複数の経路で使い、通信可能にしたものである。そのため、既存アプリケーションに変更を加えることなくそのまま使用することができる。図 1 は TCP に関する階層の図である。三層はそれぞれ左右で対応しており、複数のサブフローを作ることにより複数の経路を実現している。経路ごとに通信する割合を設定することもで

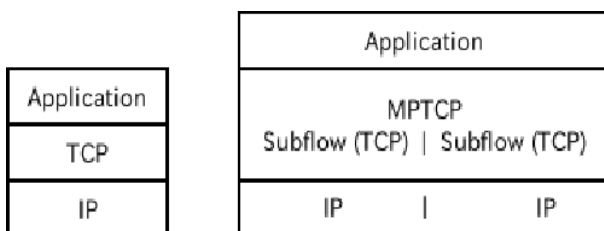


図 1 MultiPath TCP 階層図

きる。通信の開始には、MPTCP オプション 0x0 の MP_CAPABLE を 3-way handshake の際に送り、サーバ側とクライアント側で MultiPath TCP が実装されているかを確認する。双方で実装されていた場合、MultiPath TCP を用いて通信する。

3 実装状況

MultiPath TCP は Linux カーネル Version3.0 以降及び FreeBSD10.0 以降であれば、MultiPath TCP パッチをあてることにより実装することができる。[3] Linux ディストリビューション毎のパッチも用意されている。OSX10.10(Yosemite) 以降と iOS7 以降は既に実装されている。Android はエミュレートに限定されるが、一部バージョンで使用可能となるパッチが配布されている。

4 利点

MultiPath TCP を用いる一つの利点は、優先制御が可能なため通信の割合を変更し、通信速度が速い経路の割合を増やせることである。これは、経路の一部に遅延が発生している場合に有効である。

二つ目に、複数の接続をしているため、通信の断絶が起こりにくい。これは、特に携帯端末で移動中に公衆無線 LAN が複数あったり、電波が弱い場合、通常はモバイルデータ通信と Wi-fi が切り替わることにより接続が何度も切れるが MultiPath TCP であれば通信を継続できる。

5 エミュレータ間の接続方法

VirtualBox にはホストオンリーネットワークというゲスト OS とホスト OS を繋ぐネットワークを作ることができる。CORE には rj45 というエミュレータ外と通信できるツールがあるため、これらを接続する。図 2 は CORE

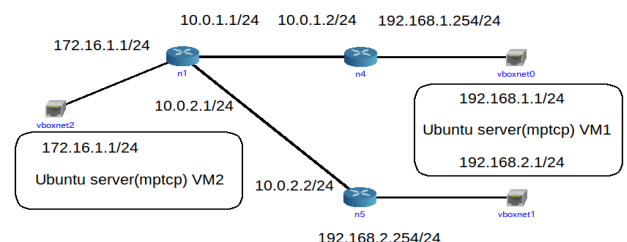


図 2 ネットワーク構成図

上のネットワーク構成図を表したものである。VM1 を Client, VM2 を Server と仮定する。VM1 にはアダプタが 2 つ、VM2 には 1 つ接続されており、それぞれ図のようにアドレスを割り振った。10.0.0.0/16 内を外部インターネットと仮定し、ここで遅延やロスを設定する。

接続の手順を一つでも間違えるとネットワークが繋がらなくなるため、以下の手順で行った。まず Virtualbox のホストオンリーネットワーク内に vboxnet をそれぞれ作成した後、CORE を起動し、エミュレートする。この vboxnet を先に作らなければ、rj45 が認識せず繋がらないためであ

る。最後に Virtualbox でゲスト OS の各アダプタをホストオンリーアダプタで設定し、ゲスト OS をエミュレートする。この時に Virtualbox を先に起動すると、rj45 と繋がらなくなる。CORE 内では動的ルーティングを用いると rj45 の先を認識しないため、Virtualbox でエミュレート後にそれぞれのルータで経路設定する。VM1 では、ゲートウェイなどの設定が通常のものではなくマルチホーミングの方法を取る。ただし、優先度はつけない。

6 実験環境と手順

図 2 のネットワーク構成で実験を行う。それぞれの VM には Ubuntu Server 14.04 を使用する。近年は、外国との通信が増えているためそれを考慮し、通常時の遅延として San Francisco との通信で発生する遅延の平均を用いた。異なる ISP を用いて通信を行っていることを仮定するため、学校から (経路 1)、自宅から (経路 2) との通信の遅延を用いた。遅延はそれぞれ $145\text{ms} \pm 1.0\text{ms}$ 、 $120\text{ms} \pm 5.0\text{ms}$ であり、帯域幅はそれぞれ 80Mbps、40Mbps とする。

計測では iperf を用い、ウィンドウサイズには 512Kb で計測した。それぞれ計測は 10 回ずつ行い、差の大きなものには 95% 信頼区間を求めた。

実験は以下の順番で行った。

1. MultiPath TCP での輻輳制御アルゴリズム比較
2. Regular TCP と MultiPath TCP の遅延比較
3. Regular TCP と MultiPath TCP のロス比較
4. 携帯端末での使用を想定した回線速度での Regular TCP と MultiPath TCP の比較
5. OS での差を調べる通常遅延の Ubuntu14.04 と Mac OS での MultiPath TCP 同士での比較
6. Regular TCP と MultiPath TCP のファイル取得速度の比較

まず、輻輳制御アルゴリズムを比較するため、通常の遅延、パケットロスなしで、client 側で cubic, reno, olia, wvegas を用いて実験した。次に、一番速かった輻輳制御を用いて、遅延をそれぞれの値でロスなしで実験し、通常の遅延でパケットロスをそれぞれの値で実験した。要旨では、これらの実験結果を記した。

7 実験結果

ロス率ごとの実験は、遅延の揺らぎも遅延と等倍に設定した。

7.1 輻輳制御比較

Olia は最大フレームワーク、wvegas は遅延に焦点を当てた、MultiPath TCP 向けの輻輳制御アルゴリズムである。

表 1 より、本実験では cubic が最も速かった。次からの実験では cubic を使用する。

表 1 輻輳制御アルゴリズム比較

cubic	reno
$16.7 \pm 0.3\text{Mbps}$	$14.4 \pm 0.1\text{Mbps}$
olia	wvegas
$14.4 \pm 0.2\text{Mbps}$	$14.4 \pm 0.2\text{Mbps}$

表 2 遅延での比較 (Mbps)

遅延 経路 1, 2	MPTCP	TCP1	TCP2
0ms, 0ms	113.0	76.0	38.3
145ms, 120ms	16.7 ± 0.3	14.8 ± 0.4	15.4 ± 0.5
217.5ms, 180ms	8.8 ± 0.1	7.5 ± 0.1	9.4 ± 0.1
290ms, 240ms	6.0 ± 0.1	5.2 ± 0.2	6.5 ± 0.1

表 3 パケットロス率での比較 (Mbps)

loss	MPTCP	TCP1	TCP2
ロスなし	16.7 ± 0.3	14.8 ± 0.4	15.4 ± 0.5
ロス率 0.5%	3.9 ± 1.1	4.8 ± 1.3	2.2 ± 1.1
ロス率 1%	3.4 ± 0.9	0.8 ± 0.3	0.3 ± 0.1

7.2 遅延、ロスを設定した比較

表 2 より、遅延が大きくなった場合、MultiPath TCP は Regular TCP の平均ほどになっている。表 3 より、MultiPath TCP は Regular TCP に比べてロスに強いことが分かった。

8 おわりに

輻輳制御プロトコルでの比較では、cubic が他の輻輳制御プロトコルよりスループットが 1.15 倍高かった。遅延での実験ではスループットが 2 経路の合計まで上がらなかった。これは 2 経路の帯域幅に大きく差があるため起こったと考えられる。ロス率での実験ではスループットが 4 倍高いため、ロスに対しては効果がみられた。

参考文献

- [1] Christoph, P. and Olivier, B.: Decoupled from IP, TCP is at last able to support multihomed hosts., *Communications of the ACM*, 2014 April, Vol. 57, No. 4, pp. 51–57 (2014).
- [2] Ford, A., Raiciu, C. and Handley, M.: TCP Extensions for Multipath Operation with Multiple Addresses (2013). <https://tools.ietf.org/html/rfc6824>.
- [3] IP Networking Lab: MultiPath TCP - Linux Kernel implementation (accessed: jan. 2016). <https://multipath-tcp.org>.
- [4] Stewart, R., Ed.: Stream Control Transmission Protocol (2007). <https://tools.ietf.org/html/rfc4960>.