

## マルチパス TCP のスループット公平性の改善

## Improvement of Throughput Fairness of Multipath TCP

泉 大地<sup>†</sup>      樽谷 優弥<sup>†</sup>      福島 行信<sup>†</sup>      横平 徳美<sup>†</sup>  
Daichi Izumi      Yuya Tarutani      Yukinobu Fukushima      Tokumi Yokohira

## 1. はじめに

近年、複数のネットワークインタフェースを持つ機器が普及している。この複数のインタフェースを同時に利用して通信効率の増加を図るマルチパス TCP が検討され、マルチパス TCP 用の輻輳制御アルゴリズムが提案されているが、公平性が十分ではなかった [1]。

本報告では、この公平性を高める方法について検討する。

## 2. 輻輳制御方式

マルチパス TCP ではパケットロスを契機に輻輳制御を行うロスベース型の LIA (Linked Increases Algorithm) や RTT (Round Trip Time) の増加を契機とする遅延ベース型の W Vegas (Weighted Vegas) 方式などが提案されているが、今回は LIA 方式を検討の対象とする。

LIA 方式では輻輳回避状態のとき、確認応答を受信するたびに式 (1) に基づいてサブフロー  $r$  の輻輳ウィンドウ  $\omega_r$  を増加させる。なお、アプリケーションが使用できる経路の集合を  $R$  としたとき、その 1 つの経路をサブフローと呼ぶ。

$$\omega_r = \omega_r + \frac{\max_{r \in R} \omega_r}{RTT_r^2} \left( \sum_{i \in R} \frac{\omega_i}{RTT_i} \right)^2 \quad (1)$$

## 3. Linux におけるスループットを左右する制御

## 3.1 局所輻輳制御

Linux ではアプリケーション間の公平性を高めるために局所輻輳制御が実装されている。局所輻輳制御は、IP 層と NIC (Network Interface Card) との間に存在する IFQ (Interface Queue) のキュー溢れを局所的な輻輳とみなして、CWR (Congestion Window Reduced) 状態へ移行し、輻輳ウィンドウを半減する。

局所輻輳制御による CWR 状態への移行は IFQ の溢れによって生じるため、送信レートが低いアプリケーションよりも送信レートが高いアプリケーションのパケットが溢れる確率が高くなる。これによって送信レートが高いアプリケーションの輻輳ウィンドウが半減し、ネットワークの帯域に空きができるため、送信レートが低いアプリケーションの輻輳ウィンドウが増加しやすくなる。この作用でアプリケーション間の公平性がよくなると考えられている。

## 3.2 TCP Small Queue

TCP の再送タイマーは、パケットが送信バッファから送信されたタイミングで計測を開始するため、通信路が輻輳した場合、パケットがキューに長時間格納されることでタ

イムアウトが発生し、パケットロスが発生したと送信側が誤検知して輻輳制御が誤作動する問題があった。これを Buffer Bloat 問題と呼ぶ。TCP ではこの問題の対策として TSQ (TCP Small Queue) と呼ばれる制御が導入された。この制御では、TCP の送信バッファに格納するパケットの量を制限することによって、キュー内での滞在時間が減少し、Buffer Bloat を抑制することができる。TSQ では式 (2) に基づいて格納するパケットの上限値を設定する。

$$\text{limit} = \min \left( 262144, \max \left( 8704, \frac{2 \times \text{MSS} \times \omega_r}{RTT_r} \times \frac{1}{1024} \right) \right) \quad (2)$$

## 4. スループット公平性の改善

## 4.1 従来の TSQ の問題点

従来の送信キュー制御では、送信キューに格納されるパケットの量が制限され、IFQ に格納されるパケットの量も制限されていた。そのため、IFQ のバッファオーバーフローによって生じる局所輻輳制御が働かない。また、式(2)にあるように、TSQ の制限値は輻輳ウィンドウで示される送信レートを基に決定される。そのため、輻輳ウィンドウが小さいアプリケーションは送信バッファに入れる量が少なくなる。これらの理由によってアプリケーション間の公平性が保てなくなっていた。

## 4.2 改善策

公平性を保つ 1 つの方法として輻輳ウィンドウが小さいアプリケーションの輻輳ウィンドウを増加させることが考えられる。そこで、本研究では試しとして式(2)で表される limit を定数値とすればどうなるかを実験した。

そうすることで輻輳ウィンドウが小さいアプリケーションも IFQ にパケットを入れることができるので、輻輳ウィンドウが増加し、帯域を確保しやすい状況を作り出せると考えたためである。

この送信キューに格納できるデータ量は式(3)のように従来のデフォルト値の上限値である 262144 の半分に設定する。

$$\text{limit} = 131072 \quad (3)$$

## 5. 実験環境及び実験結果

本実験では図 1 の実験ネットワークを用いる。マルチパス TCP のバージョン 0.92 を実装した Sender, Receiver と、ネットワークエミュレータを導入し、ネットワークの RTT 値やロス率を変更することができる NE1, NE2, NE3 を用いる。

<sup>†</sup> 岡山大学 Okayama University

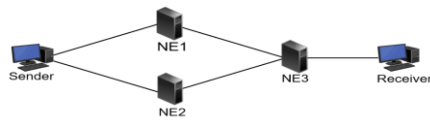


図 1. 実験環境

Sender で 3 つのスループット測定用のアプリケーションの起動時間差を 0.5 秒つけて実行する。起動した順に APP1, APP2, APP3 とする。

従来法と改善策の packet loss が無い場合のスループットを図 2 に示す。従来法では先に起動した APP1 のスループットが高くなっているが、改善策ではすべてのアプリケーションが同量のスループットとなっている。IFQ サイズが 1000 パケットのときの輻輳ウィンドウの変化を図 3 に示す。packet loss がなく、輻輳ウィンドウが最後まで変わらないので、3 秒後までの値を示している。従来法では後から起動するアプリケーションの輻輳ウィンドウが 200 程度と低くなっているのに対して、改善策では 400 と倍近くに増えている。従来法では TSQ の制限値が輻輳ウィンドウに比例していたため、初めに輻輳ウィンドウを増加することができた APP1 が送信キューに多数パケットを入れることができるためスループットが高く、後から起動したアプ

リケーションは輻輳ウィンドウが小さいため TSQ の制限値も小さくなり輻輳ウィンドウを増加させることができず、低いスループットになったと考えられる。改善策では TSQ の制限値が輻輳ウィンドウに比例せず一定のため、輻輳ウィンドウを増加させることができたと考えられる。

図 4 に packet loss がある場合のスループットを示す。packet loss がある場合はない場合と比較してアプリケーション間の値に多少の違いがある。これは、packet loss が確率的に生じるため、packet loss の回数やタイミングによって輻輳ウィンドウに影響し、スループットが変化するからである。

## 6. むすび

今回の実験では、TSQ の制限値を変更することでアプリケーション間の公平性を向上させることができる可能性を示した。しかし、制限値を固定すると違う環境では結果が異なる可能性がある。そこで、今後の課題は TSQ の制限値を動的に変化させ、スループットが公平となるようにできる機構を考案することである。

### 参考文献

- [1] 刈谷 騎士, 福島 行信, 横平 徳美, “マルチパス TCP における輻輳制御方式のスループット公平性の検討”, 第 69 回 電気・情報関連学会中国支部連合大会, (2018).

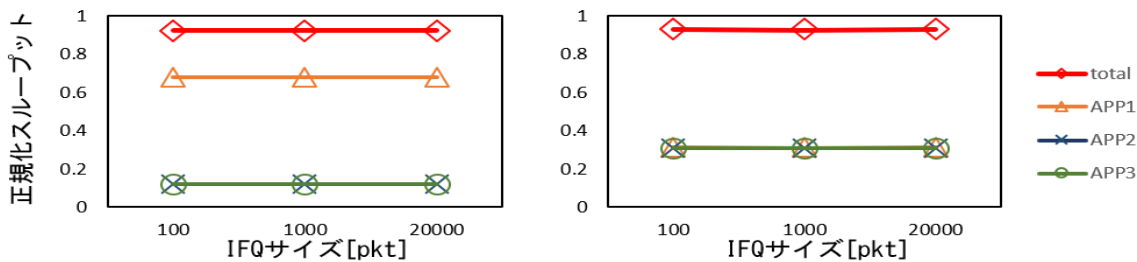


図 2. RTT=50ms, ロス率 0%, (左) 従来法 (右) 改善策

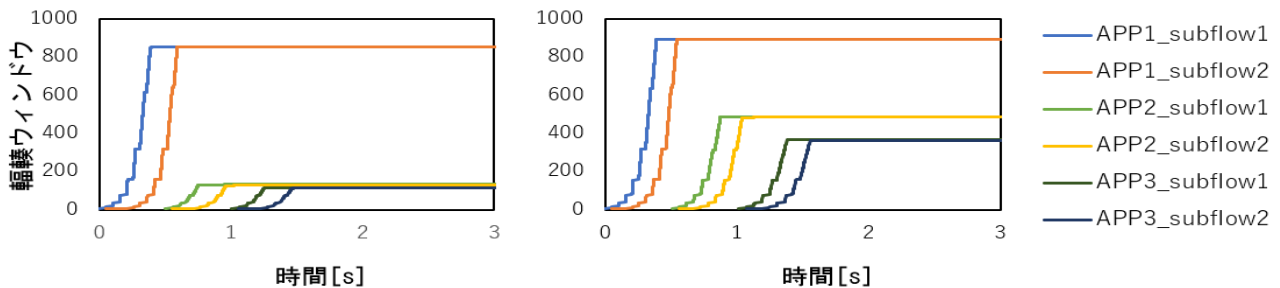


図 3. RTT=50ms, ロス率 0%, IFQ=1000pkt, (左) 従来法 (右) 改善策

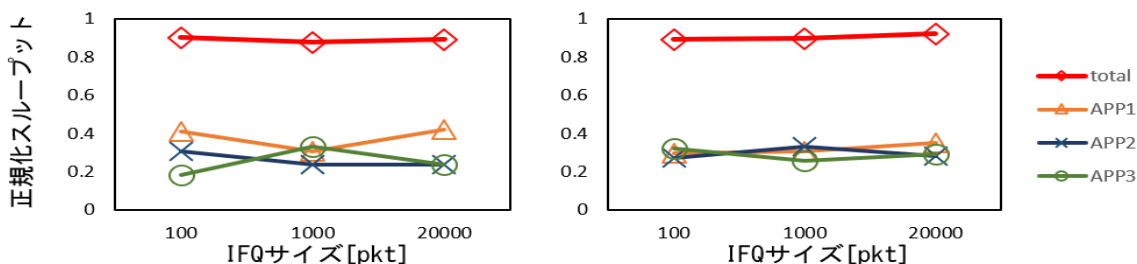


図 4. RTT=50ms, ロス率 0.001%, (左) 従来法 (右) 改善策