

BBR と CUBIC の共存時における性能評価

屋代 秀幸[†] 石 為之[‡] 内海 哲史[‡][†] 福島大学理工学群共生システム理工学類 [‡] 重慶理工大学電気電子工程学院

1 はじめに

Linux カーネル 2. 6. 19 以降で TCP CUBIC がデフォルトの TCP 輻輳制御アルゴリズムとして用いられている。TCP 輻輳制御アルゴリズムは、主にパケットロスに基づいて輻輳を検出する Loss-based 輻輳制御と、遅延時間に基づいて輻輳を検出する Delay-based 輻輳制御がある。TCP CUBIC は Loss-based 輻輳制御であり、いくつかの問題点が挙げられる。

バッファが大きいときには、過剰なバッファリングにより Bufferbloat[1] が発生し、無線リンクでは、リンクエラーによるパケット損失が起きやすく、スループットが低下してしまう。

本稿では、Google 社が 2016 年 9 月に発表した輻輳制御アルゴリズム TCP BBR[2] に注目した。TCP BBR はネットワークにおけるデータ転送レートと RTT(Round Trip Time) を計測し、最大のデータ転送レート (BtlBW), 最小の RTT (RTprop) から BDP (Bandwidth-Delay Product) を計算し、バッファが埋まらないようにする。本稿では、これから普及することが考えられる TCP BBR と現在普及している TCP CUBIC が共存したときの性能について実験及び考察をした。

2 TCP BBR

TCP BBR は、Linux カーネル 4. 9 以降で利用可能となった輻輳に基づく輻輳制御である。実際の輻輳に反応することにより、ネットワーク利用率を最大化し、キュー長を最小に保つ。

TCP BBR は、RTT からキューイング遅延を抜いた往復伝搬遅延 (RTprop) と、ボトルネックのリンク容

量 (BtlBW) を推定し、RTprop と BtlBW の積である帯域遅延積 (BDP) を計算し、経路にあるパケットを BDP と同量にすることにより、キュー長が短い状態で高スループット・低レイテンシを実現する。

ACK を受け取るごとに RTT を計測し、過去数十秒間に計測された複数の RTT の中で最小の値を RTprop の推定値とする。ある時刻に送信したパケットについて、その ACK が返ってくるまでに受け取った ACK パケットの合計を時間で割った値をデータ転送レートの推定値、過去の一定期間におけるデータ転送レートの最大値を BtlBW の推定値とする。

BBR の輻輳制御アルゴリズムでは、Startup, Drain, ProbeBW, ProbeRTT の 4 つの状態を遷移する。安定すると、ProbeBW と ProbeRTT の 2 つのモードを繰り返す定常状態となる。

2.1 Startup モード

スロースタート同様に指数関数的に送信レートを増加させ、通信経路を一杯にすることを目標とする。キューが上限に達したら Drain モードへと遷移する。

2.2 Drain モード

一杯になったバッファからパケットを減らすことを目標とする。バッファが空になると定常状態である ProbeBW モードへと遷移する。

2.3 ProbeBW モード

リンク容量をフルにしつつ、バッファが短い状態を保つ。サイクルフェーズと呼ばれるものに入り、送信レートを増やしたり減らしたりする。

2.4 ProbeRTT モード

RTprop が 10 秒以上更新されないとき、このモードに入る。ウィンドウサイズを最小にして 200ms 程度の間、通信速度を強制的に落とす。

3 実験環境

実験をするにあたり、Linux マシン (Ubuntu17. 04, Linux カーネル 4. 12. 4) 2 台と FreeBSD マシンを 2

Co-existence Problem of BBR and CUBIC in Deep Buffer Link

Hideyuki YASHIRO, Shi WEIZHI and Satoshi UTSUMI

[†]Fukushima University, Japan

[‡]Chongqing University of Technology, China

台用意した。

4 実験方法

TCP BBR 単一の場合について、Linux マシン 1 台から TCP BBR でデータを送信しスループットの測定を行う。Linux マシン 2 台からデータを同時に送信し、TCP BBR フローと TCP CUBIC フローが同一経路で共存するとき (TCP BBR vs. TCP CUBIC) のスループットを測定する。バッファの大きさを BDP の 1~18 倍とし、18 回の測定を 1 セットとする。

実験で用いる TCP のパラメータを表 1, 2 に示す。

表 1 送信側のパラメータ

最大ウィンドウサイズ	100MBytes
データ送信時間	120 秒

表 2 ボトルネックリンクのパラメータ

リンク容量	10Mbps
バッファサイズ	1~18(BDP)
RTprop	40ms

5 実験結果

TCP BBR のスループット測定結果を図 1 に、TCP BBR vs. TCP CUBIC の 10 回測定したスループットの平均値を図 2 に示す。

TCP BBR vs. CUBIC のグラフのように、バッファが小さいときは TCP BBR は高性能で TCP CUBIC は低い性能であった。バッファを大きくしていくにつれ TCP BBR の性能は下がり、TCP CUBIC の方が高い性能となる。tcptrace を用いて解析を行ったところ、スループットの合計が下がっていく原因は、連続的なタイムアウトが起きていることによると分かった。

6 まとめ

従来のパケットロスに基づく輻輳制御と異なり、輻輳に基づく輻輳制御である TCP BBR は、バッファの大きさに左右されず安定して高い性能を維持することができる。しかし、既存の TCP 輻輳制御アルゴリズムである TCP CUBIC と共存したとき、バッファが大きくなると性能が低下するという問題があることが分かった。

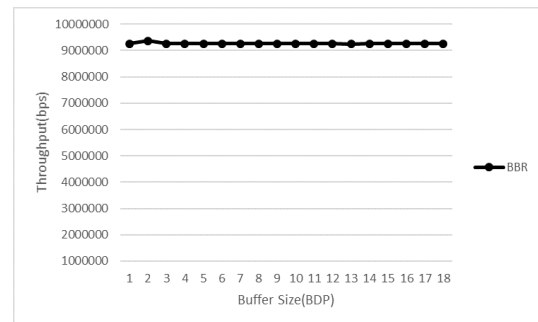


図 1 TCP BBR のスループット測定

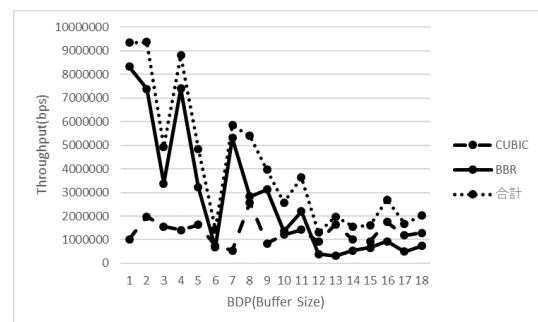


図 2 TCP BBR vs. TCP CUBIC のスループット測定

参考文献

- [1] Vint Cerf, Van Jacobson, Nick Weaver, Jim Gettys, "BufferBloat: What's Wrong with the Internet?", Magazine Queue - Log Analysis, Volume 9, Issue 12, December 2011.
- [2] Neal Cardwell, Yuchung Cheng, C. Stephen Gunn, Soheil Hassas Yeganeh, Van Jacobson, "BBR: Congestion-Based Congestion Control," ACMqueue, Volume 14, Issue 5, December 2016.