

L-009

RTT を用いた輻輳制御 RTT-Based Congestion Avoidance

告 大輔[▲] 武田 利浩[▲] 平中 幸雄[▲]
Daisuke Tsuge Toshihiro Taketa Yukio Hiranaka

1. はじめに

今日、インターネット利用者の増加と、動画等リッチコンテンツの人気の進んでいる。それに伴い、回線の混雑により発生する「輻輳」をどのように扱うかが重要視されてきている。

今現在広く使用されている方式では、輻輳を回避することができない。そこで考案されたのが「RTT(Round Trip Time)」の増加から輻輳を予測し、回避することを目的とする DCA(Delay-based Congestion Avoidance)アルゴリズムである。

本研究では、DCA アルゴリズムの特性と課題を明らかにし、新たな手法の DCA を提案することを目的とする。

2. 関連技術

2.1 TCP Reno^[1]

TCP Reno は、今現在最も多く利用されている方式であり、そのアルゴリズムは式(1)の通りである。

$$cwnd(t') = \begin{cases} cwnd(t) + \frac{1}{cwnd(t)} \dots \text{パケットロス無し} \\ \frac{cwnd(t)}{2} \dots \text{パケットロス発生} \end{cases} \quad (1)$$

cwnd : ウィンドウサイズ(パケット単位)

輻輳によりパケットロスが発生するまで、ウィンドウサイズを増加させるが、パケットロスが発生した際には現在のウィンドウサイズを半分減らす。このアルゴリズムは AIMD(Additive Increase Multiplicative Decrease)アルゴリズムと呼ばれる。

2.2 TCP Vegas^[2]

TCP Vegas は、RTT を用いて輻輳予測を行う手法として初めて提案された方式である。そのアルゴリズムを式(2)に示す。

$$cwnd(t') = \begin{cases} cwnd(t) + 1 \dots Diff \times baseRTT < \alpha \\ cwnd(t) \dots \alpha < Diff \times baseRTT < \beta \\ cwnd(t) - 1 \dots \beta < Diff \times baseRTT \end{cases} \quad (2)$$

α, β : 定数
baseRTT : 今まで計測された最小の RTT
Diff = *Expected* - *Actual*
Expected(期待値) = *cwnd* / *baseRTT*
Actual(実測値) = *cwnd* / *RTT*

最小の RTT を用いて計算した期待値から現在の RTT から求めた実測値の差を取ったものと、定数 α, β の大小関係によってウィンドウサイズの変更を行う。このように RTT を元に輻輳制御を行うアルゴリズムを DCA アルゴリズムと言う。

2.3 FAST TCP^[3]

FAST TCP は、TCP Vegas の抱える問題点^[4]を改善したもので、TCP Vegas の高速バージョンと言える。そのアルゴリズムは式(3)の通りである。

$$w \leftarrow \min \left\{ 2w, (1-\gamma)w + \gamma \left(\frac{baseRTT}{RTT} w + \alpha \right) \right\} \quad (3)$$

w: ウィンドウサイズ
 α : 定数
 γ : 0~1 の間の定数

また、 $2w$ にはならないとすると式(3)のようにも表せる。

$$w_i(t+1) = w_i(t) + \gamma(\alpha - x_i(t)q_i(t)) \quad (3)'$$

$w_i(t)$: 最新の *t* 中のフロー *i* のウィンドウサイズ
 $q_i(t)$: 現在のキュー時間 (*RTT*-*baseRTT*)
 $x_i(t)$: 現在のスループット (*w*/*RTT*)

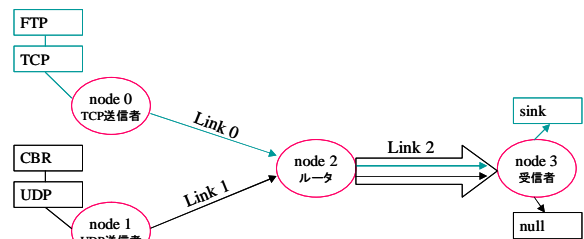
FAST TCP では、TCP Vegas のようにウィンドウサイズを 1 ずつではなく、期待値と測定値の差の大きさによって、増減させる大きさを変えている。

3. 確認実験

ns-2 を用いて、上記 3 つの TCP の振る舞いを確認するためにシミュレーション実験を行った。

3.1 ネットワークトポロジ

図 1 のトポロジを用いて、Node 0 に TCP Reno, TCP Vegas, FAST TCP を用い、計 3 回の実験を行った。



Link 0, Link 1...帯域幅 1Mbps 遅延 2ms
 Link 2...帯域幅 1.5Mbps 遅延 5ms
 ルータのキューサイズ...20

図 1: ネットワークトポロジ

3.2 シナリオ

シミュレーションのシナリオは以下の通りである。

- シミュレーション開始後1秒にTCPを開始
- 15秒後にUDPを開始
- 45秒後にUDPを停止
- 60秒後にTCPを停止

3.3 結果

シミュレーションの結果を図2に示す。

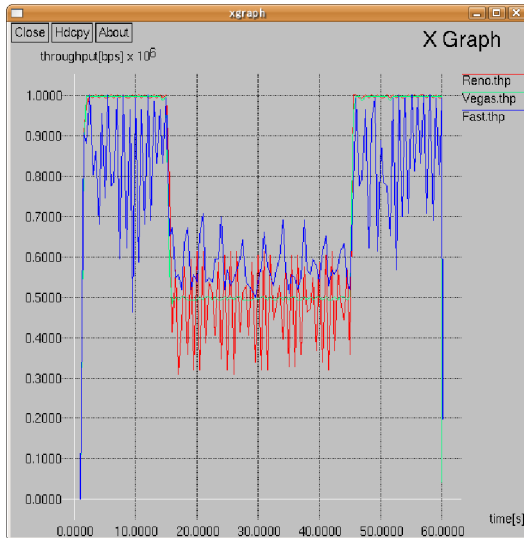


図2:シミュレーション結果

また、各時間帯での平均スループットは表1のようになった。

表1:各時間帯での平均スループット

	0s~15s	15s~45s	45s~60s	全体
TCP Reno	926Kbps	491Kbps	954Kbps	718Kbps
TCP Vegas	925Kbps	508Kbps	951Kbps	725Kbps
FAST TCP	780Kbps	587Kbps	817Kbps	694Kbps

3.4 考察

競合するフローが存在しない場合(0s~15s, 45~60s)には、TCP Reno が一番高いスループットを実現した。これは、AIMD アルゴリズムによる積極的なウィンドウサイズの増加と、競合フローが無いためにパケットロスが起こりにくかったことが理由と思われる。

一方、競合するフローが存在する場合(15s~45s)にはFAST TCP が一番高いスループットを実現した。このことから、FAST TCP の輻輳予測が効果的に機能していることがわかる。しかし、競合するフローが存在しない場合においてFAST TCPのスループットが伸び悩んだ。図2からFAST TCPは、上限の1Mbps付近までスループットが上昇したのち、直ちにスループットが低下していることが分かる。このことから、 $baseRTT$ と観測した RTT が近い状態(1Mbps 付近)の際に、ウィンドウサイズを上げすぎ、結果 RTT が増加し、ウィンドウサイズの減少が起こり、スループットが低下していると思われる。

4. 提案方式

シミュレーション結果から、FAST TCP の性能を改善するためには、 $baseRTT$ と測定した RTT が近い場合にウィンドウサイズの変化量を少なくすることが考えられる。そこで本研究では、式(3)のうち、ウィンドウサイズの増減に関係する部分 $\gamma(\alpha - x_i(t)q_i(t))$ に着目し、どの程度差を反映するか決めていたパラメータ γ の値を、 $baseRTT$ と測定した RTT が近い場合に小さくする方法(方式1)と、変化させるウィンドウサイズを決めている部分 $(\alpha - x_i(t)q_i(t))$ を乗じることで、差の絶対値が1未満の場合は変化を抑える方法(方式2)の2つを提案する。

◆方式1

$$w_i(t+1) = w_i(t) + \gamma'(\alpha - x_i(t)q_i(t)) \quad (4)$$

式(3)のパラメータ γ を、 $baseRTT$ と観測した RTT の差の絶対値によって動的に変化させることにより、 $baseRTT$ と測定した RTT が近い場合には積極的な変化を避ける。

◆方式2

$$w_i(t+1) = \begin{cases} w_i(t) + \gamma(\alpha - x_i(t)q_i(t))^n & (5.1) \\ w_i(t) + \text{sign}(\alpha - x_i(t)q_i(t))\gamma(\alpha - x_i(t)q_i(t))^n & (5.2) \end{cases}$$

n : 2以上の整数

$(\alpha - x_i(t)q_i(t))$ の部分を n 乗することにより、差が1未満の場合には変化量を抑え、逆に差が大きければ大きいほど、変化量を増やすことができる。 n が奇数の場合は式(5.1)を、偶数の場合は式(5.2)を適用する。

5. まとめ

TCP Reno, TCP Vegas, FAST TCP についてシミュレーション実験を行い、FAST TCP が抱える問題を発見し、その問題を解決すべく2つの方法を考案した。

今後は、考案した方式を ns-2 に実装し比較実験を行うと共に、帯域幅等のパラメータを変化させながらの測定実験を行って行きたい。

参考文献

- [1] Dah-Ming CHIU, Raj JAIN, Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks, Computer Networks and ISDN Systems, p.p.1-14, 1989.
- [2] Lawrence S. Brakmo, Sean W. O'Malley, Larry L. Peterson, TCP Vegas: New Techniques for Congestion Detection and Avoidance, ACM SIGCOMM Computer Communication Review Vol.24 Issue 4 p.p.24-35, Oct. 1994.
- [3] Cheng Jin, David Wei, Steven H. Low, Julian Bunn, Hyojeong D. Choe, John C. Doyle, Harvey Newman, Sylvain Ravot, Suresh Singh, Caltech Fernando Paganini, Gary Buhrmaster, Les Cottrell, Olivier Martin, Wuchun Feng, FAST TCP: from theory to experiments, IEEE Network, Vol.19, Issue:1 Jan.-Feb. 2005.
- [4] Jim Martin, Arne Nilsson, Injong Rhee, The incremental deployability of RTT-based congestion avoidance for high speed TCP Internet connections, Joint International Conference on Measurement and Modeling of Computer Systems Proceedings of the 2000 ACM SIGMETRICS international conference on Measurement and modeling of computer systems, p.p.133-144, 2000.