

フロー制御におけるTCP輻輳ウィンドウサイズ削減率について On Decrease Ratio of TCP Congestion Window Size for Cross-layer Flow Control

上川原大治[†] 工藤智子[‡] 武田利浩[‡] 平中幸雄[‡]
Daiji Kamikawara Tomoko Kudo Toshihiro Taketa Yukio Hiranaka

1. はじめに

TCPの輻輳制御は、パケットロスを検出してから動作するため、帯域の利用効率が低下する[1]。伝送往復時間を推測して制御する方法[2]も提案されているが、本研究では、直接的なフロー制御により輻輳制御問題を解決する手法としてクロスレイヤー通信を用いる[3]。ネットワーク中継機器からの情報通知によりTCP輻輳ウィンドウサイズを削減し、輻輳状態を改善する。本研究報告では、適切な制御を実現する輻輳ウィンドウサイズの削減率を求め、その制御効果をシミュレータを用いて検証した。

2. クロスレイヤー通信によるTCP輻輳制御概要

提案手法は、中継機器のバッファ利用率が基準値(通知しきい値)に達した際、TCP層にクロスレイヤー通信で通知することで輻輳状態を改善する。通知を受けた送信元TCPが輻輳ウィンドウサイズ(cwnd:Congestion Window Size)を下げることで輻輳制御を行う。また、本手法は、一般的に利用されるTCP Renoに制御を追加している。

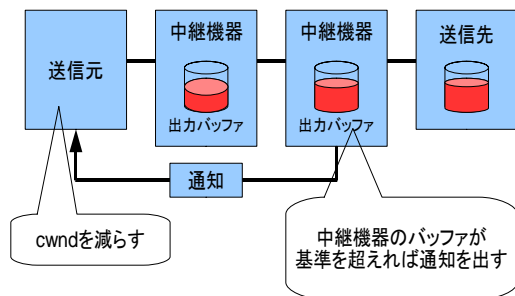


図1 クロスレイヤーフロー制御の概要

中継機器からの通知により、cwndの削減率をネットワークの状態を反映した値になるように設定する。ここで、通知を受けた際のcwnd削減率を β と表す。このとき、中継ノードからの輻輳通知を受けたTCPはcwndを以下の式に従うように変更する。

$$cwnd_{New} = \beta \times cwnd_{Old} \quad (0 < \beta < 1) \dots (1)$$

β はフローごとに計算される。N個のフローがボトルネックリンクを通るとき、k番目のフローのcwnd削減率 β_k を式(2)で与える。

$$\beta_k = (Bw - \gamma \frac{Q_B}{RTT_k}) / T \dots (2)$$

Bw : ボトルネックリンク帯域幅(bps)

Q_B : 輻輳通知時バッファ量(bit)

T_i : フローiの送信帯域幅(bps)

T : ボトルネック通過フローの合計送信帯域幅(bps)

RTT_k : k番目のフローの往復伝送遅延時間(s)

γ : 制御パラメータ ($0 < \gamma < 1$)

通知にかかる時間による誤差をないものとし、受信ノードから告知されるウィンドウサイズが十分大きいとする、

$$T_k = \frac{cwnd_{old}}{RTT_k} \dots (3)$$

と表せる。この時、k番目のTCPフローのcwnd更新式は、式(1)(2)(3)より

$$cwnd_{New} = Bw (RTT_k - \frac{\gamma Q_B}{Bw}) \frac{T_k}{T} \dots (4)$$

となる。式(4)は、「ボトルネックリンクの帯域幅と往復遅延の積」が各フロー単独での最適cwndであるが、往復遅延から「キュー長相当分の遅延時間」を差し引き、「合計送信帯域幅に占める自身の送信帯域幅の割合」で比例配分したcwndを計算することを意味している。 γ の値はキュー長分の差し引き量を低減し、キューを空にしない程度を決める係数である。 $0.1 < \gamma < 1$ の範囲でボトルネック帯域の利用率は良好であったが、以下に示すシミュレーションでは、 $\gamma=0.5$ とした結果を示す。また、受信側が告知するウィンドウサイズは、十分大きなものとする。

3. シミュレーション方法

ネットワークシミュレータ ns-2[4]を用いて提案方法を検証する。図2に示すネットワークトポロジで、シミュレーションを行った。

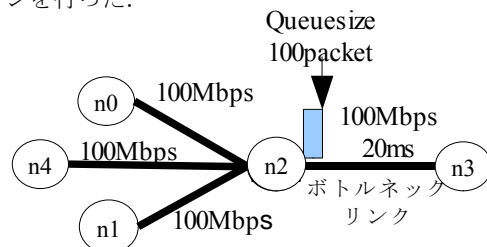


図2 ネットワークトポロジ

シミュレーションの評価は、ボトルネックリンクにおける平均トラフィック量とし、リンク帯域幅100Mbpsに近いほど、ボトルネックリンクの利用率が高いと判断する。

[†] 東北インフォメーション・システムズ(株) Tohoku Information Systems Co. Inc.

[‡] 山形大学 Yamagata University

• 山形県立産業技術短期大学校 Yamagata College of Industry and Technology

ノード n0, n1, n4 からノード n3 へ TCP フローの送信を行う。通信開始時間は、ノード n0 がシミュレーション開始と同時に、その後 1sec 間隔で n1, n4 がそれぞれ送信を開始する。50sec 後にすべての通信を終了する。また、TCP が送信するパケットサイズを 1KB とする。

表 1 に示す n0, n1, n4 からノード n2 間の遅延条件(a),(b),(c),(d)に対して、本提案手法による制御を実現した場合と、制御なしの場合とを比較する。遅延条件ごとにバッファ利用率の基準値(通知しきい値)を最大キューサイズに対して 10%刻みで検証する。

表 1 各ノード間の遅延時間[ms]

条件	n0-n2	n1-n2	n4-n2
(a)	1	1	1
(b)	10	10	10
(c)	100	100	100
(d)	10	20	30

4. 結果

図 3 に、シミュレーション結果を示す。本提案方式による制御あり(しきい値 20%-90%)、制御なしの場合におけるボトルネックリンクの平均トラフィック量を示した。

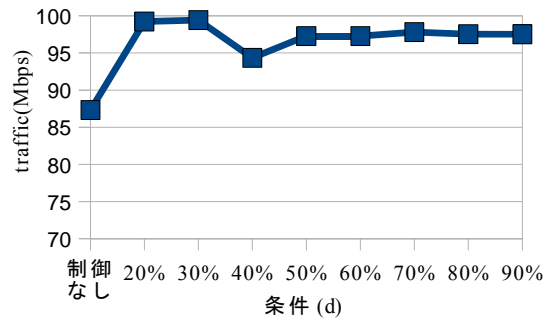
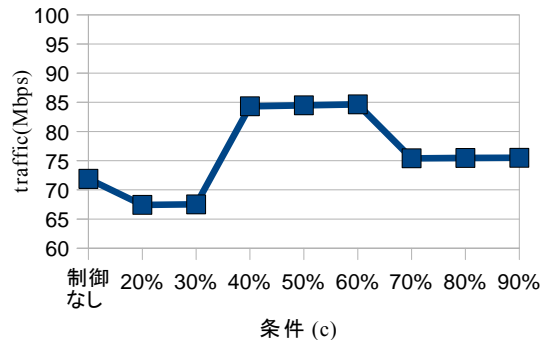
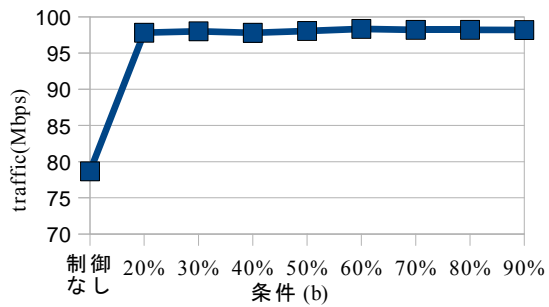
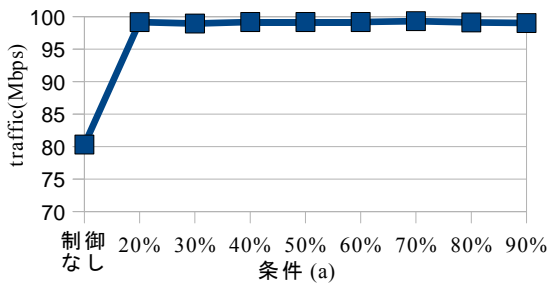


図 3 シミュレーション結果

条件(c)の通知しきい値 20-30%の場合を除き、制御なしと比較して、本提案手法による制御手法を用いた方が使用帯域幅が向上し、制御が有効であることがわかった。条件(a),(b),(d)では、制御をすることで帯域を上限の 100Mbps 近くまで使い切ることができている。また、条件(a),(b)では通知のしきい値による結果の変動が少ない。条件(d)ではしきい値条件による変動がわずかにみられる。

条件(c)では制御をすることで使用帯域幅の値が悪化してしまうことがあった。さらに通知しきい値による使用帯域幅の変動も非常に大きいという結果になった。原因は、転送開始から使用帯域が安定するまでに時間がかかってしまったからだと推測される。これは TCP Reno の cwnd を上昇させる速度が遅いという問題と、送信開始直後は使用帯域幅が安定しないという二つが原因だと考えられる。

5. まとめと今後の課題

本研究報告では、輻輳状態における帯域利用効率の低下を、中継機器からの直接通信によって改善する方法について検討した。特に、適切な制御を実現するために、輻輳ウィンドウサイズの削減率を求め、その制御効果をシミュレータを用いて検証した。その結果、遅延が大きく、通知しきい値が小さい場合以外は、本提案手法による制御手法を用いた場合の方が使用帯域幅が向上し、制御が有効であることがわかった。本提案手法では、輻輳ウィンドウサイズの削減による制御を行ったが、輻輳の緩和時や、遅延時間が大きい通信に対して、ウィンドウサイズを上げる制御方法を追加する必要がある。

参考文献

[1] 鶴正人 他, “長距離高速通信のための TCP 性能改善技術の動向”, 情報処理, Vol.44, No.9, pp.951-957, 2003.
 [2] D. X. Wei, C. Jin and S. H. Low, “FAST TCP: Motivation Architecture, Algorithms, Performance,” IEEE /ACM Transactions on Networking 14(6):1246-1259, 2006.
 [3] 高橋淳, 平中幸雄, 武田利浩, “バッファ情報を用いるクロスレイヤフロー制御”, FIT2010 (第9回情報科学技術フォーラム), 第4分冊, L-010, pp.185-186, 2010.
 [4] The Network Simulator ns-2 <http://www.isi.edu/nsnam/ns>