広域通信のためのTCPの高速化に関する一検討

6H－1

三宅 優† 神崎 昭浩‡ 加藤 聰彦† 鈴木 健二†

国際電信電話(株) 研究所† 電気通信大学‡

1. はじめに

ATM網に代表される広帯域広域網の導入に伴い、遠隔の計算機の間で高速なTCP/IP通信が可能となっている。ftpやWWWで使用されているTCPでは、フロー制御のためのウィンドウサイズの制限や、輻輳回避アルゴリズムに起因するデータ転送レートの低下などにより、特に国際ATM網のような伝送遅延の大きい広帯域網において、スループットが制限されるという問題点が指摘されている。これに対応するために、ウィンドウサイズを拡大するウィンドウスケールオプションや、輻輳などにより紛失したTCPセグメントのみを再送する選択確認(SACK)オプションが導入されようとしている。しかし、これらのオプションを使用した場合でも、従来の輻輳回避アルゴリズムが採用されており、ネットワークの帯域をより有効に使用するためには新たなアルゴリズムが必要であると考えられる。これに対し筆者らは、上記の2つのオプションを使用した場合に、SACKオプションの効率的な再送機能を前提として、輻輳時のデータ転送レートを従来よりも増加させるアルゴリズムについて検討した。本稿ではその結果について述べる。

2. 現状のTCPの輻輳制御の概要とその問題点

TCPにおいては、送信側が網の輻輳状態に合わせてデータ転送レートを制御するために、輻輳ウィンドウサイズ(以下変数cwndを用い、値はセグメント単位とする)が定義されている。セグメントが紛失し再送が発生すると、送信側は網に輻輳が発生したと判断し、cwndの値を縮小させる。タイムアウトによる再送の場合は、その時点のcwndの$\frac{1}{2}$を変数ssthreshに格納し、cwndを1に縮小させる。その後、ACKセグメントを受信する毎に、cwndを1ずつ増加させる(Slow Start)。さらに、cwndの値がssthreshを越えると、ACKセグメント毎のcwndを次式により増加する(Congestion Avoidance)。

$$cwnd \leftarrow cwnd + \frac{1}{cwnd} \qquad (1)$$

一方、同一のACKセグメントを4つ連続に受信した場合は、そのACKに対応するデータセグメントを再送し、さらに、その時点のcwndの$\frac{1}{2}$を新たなcwndとしCongestion Avoidanceの処理を行なう。

これらのアルゴリズムにより、cwndの値は、Slow Startにおいては、1セグメント分から指数関数的に、Congestion Avoidanceにおいては、再送時のcwndの$\frac{1}{2}$から線形に、それぞれ増加する。遅延の大きな広帯域広域網において大きなウィンドウサイズを使用する場合は、通常はcwndの値も大きくなる。そこでSlow StartまたはCongestion Avoidanceが生ずると、cwndが縮小し、スループットが低下するわけである。SACKオプションが使用された場合は、紛失したセグメントのみの再送と、Congestion Avoidanceによる輻輳回避が可能となり、Slow Startの起動が少なくなる。しかし、ここでもcwndの増加が線形であることがスループットの低下につながる。

3. 新しい輻輳回避アルゴリズムの提案

3.1 方針

上記のような問題点を解決するためには、cwndをなるべく長時間、最適値と思われる値に近い値に維持することが必要となる。そのためのアプローチとして、以下の2通りが考えられる。

(1) cwndが$\frac{1}{2}$に縮小した後に、最適値と考えられる付近まで急速に拡大させ、その値を保持する。

(2) 最適値を保持している場合には、必要以上にcwndを縮小させない。

既存のTCPでは、輻輳発生時に送出するデータ量を減らす方針を採用しているため、(2)のアプローチを採用すると、輻輳発生時には他のTCPトラヒックのみがSlow StartやCongestion Avoidanceを繰り返し、著しい不公平が生ずることになる。このため、(1)のアプローチを採用することとし、輻輳を検出すると従来のようにcwndを縮小させ、その後Congestion Avoidanceにおいて、cwndを最適値と考えられる値まで急速に拡大するアルゴリズムを検討することとした。この時、SACKオプションを使用し、急速にcwndを拡大させて、再び輻輳によるセグメント紛失が生じても、効率的に再送可能であることを前提とする。

3.2 アルゴリズム

(1) 直前の再送時におけるcwndの値を、cwndの最適値と見なすこととし、その値を格納する変数c_peakを導入する。変数c_peakに格納された値は、次に再送が発生して更新されるまで、保持される。

"A Study on High Speed TCP Communication over Wide Area Network"

Yutaka MIYAKE†, Akihiro KANZAKI‡, Toshihiko KATO† and Kenji SUZUKI†

KDD R & D Laboratories†, The University of Electro-Communications‡

(2) Congestion Avoidance における cwnd の値の更新を、式 (1) の代りに、以下の式を用いて行なう。

$$cwnd \leftarrow cwnd + \underbrace{\frac{|cwnd - \frac{c\_peak}{2}|}{cwnd}}_{A} \times \underbrace{\frac{|c\_peak - cwnd|}{cwnd}}_{B} \times \underbrace{\frac{\frac{c\_peak}{2}}{cwnd}}_{C} + \underbrace{\frac{1}{cwnd}}_{D} \quad (2)$$

この式において A 項から C 項の積が、Congestion Avoidance と比べて、cwnd の拡大の速度を上げる部分である。これらの各項は以下のような意味を持つ。A 項は、Congestion Avoidance の開始時における急な拡大を防ぐことを目的としている。B 項は、cwnd の最適値 (c_peak の値) の付近で、cwnd の値の拡大を緩めるために導入した。さらに C 項は、cwnd の拡大率を調整するための係数である。最後に、D 項は、既存の Congestion Avoidance と同様である。ここでは、Congestion Avoidance の開始時および c_peak の値に等しくなった時点で、cwnd の増加量が 0 となることを防いでいる。

4. 性能評価

提案したアルゴリズムを評価するために、ネットワークシミュレータ ns[1] を用いたスループット測定を行なった。評価には、図 1 に示す、ルータ間に伝送遅延の大きなボトルネックリンクがあるネットワーク構成を用いて、ホスト間で 30 秒間のデータ転送を行なった。この際、セグメントサイズは 4096 バイト、ルータのバッファは 50 セグメント分 (200K バイト) とし、ウィンドウサイズ (cwnd の最大値: maxcwnd) を変化させた。

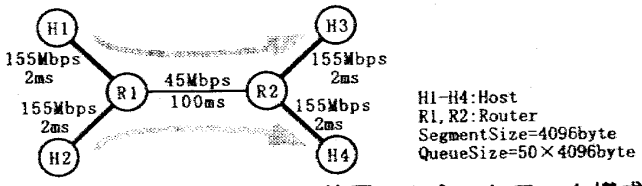

図 1: シミュレーションに使用したネットワーク構成

まず、ホスト H1 と H3 の間で単一の TCP トラヒックのみが存在する場合について、既存の SACK オプションを用いた TCP(SACK-TCP) および提案したアルゴリズムに対してスループットを測定した。その結果を表 1 に示す。この結果、maxcwnd がルータのバッファを越えた場合は、提案アルゴリズムの方が高いスループットを得た。また、maxcwnd が 512K バイトの場合の cwnd の時間的変動を図 2 に示す。この図により、提案アルゴリズムでは、Congestion Avoidance 直後は緩やかに拡大しはじめるが、直ぐに拡大率が増大し、c_peak 付近で一旦緩やかになり、さらにそれを越えると急速に増加するという変動を示していることが分かる。これは、既存の SACK-TCP の cwnd の線形な拡大と対照的である。この変動の積分値が転送したデータ量となるため、提案アルゴリズムの方がスループットが高くなる。

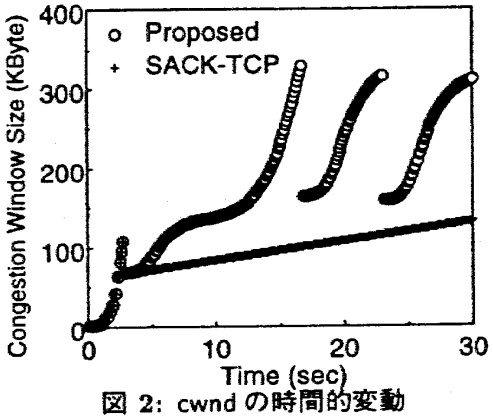

図 2: cwnd の時間的変動

表 1: 単独の TCP トラヒックのスループット

maxcwnd [K バイト]	Throughput[Mbps] SACK	提案
1024	23.5	32.6
512	23.5	32.6
256	8.3	21.4
128	17.8	17.8

表 2: 2 つの TCP トラヒックのスループット

maxcwnd [K バイト]	Throughput[Mbps] SACK×2	提案 ×2	提案	SACK
512	35.1	33.4	24.7	9.7
256	20.9	24.8	13.0	9.8
128	30.3	30.3	17.7	12.6

次に、2 対のホストの間で TCP トラヒックを発生させた場合のスループットを表 2 に示す。双方が SACK-TCP の場合と、双方が提案アルゴリズムの場合の合計のスループットは、ほぼ同程度の高い値となり、いずれの方式も有効であるという結果を得た。また、提案アルゴリズムと SACK-TCP を混在させた場合は、提案アルゴリズムの方が高いスループットとなった。しかし、SACK-TCP も一定のスループットを達成しており、提案アルゴリズムが極端に既存 TCP のスループットを低下させないことも明らかとなった。

5. まとめ

本稿では、広帯域の広域網における TCP 通信に対して、高いスループットを得るための輻輳回避アルゴリズムを提案し、シミュレーションによる性能評価を行なった結果を示した。既存の TCP では Congestion Avoidance において輻輳ウィンドウを線形に拡大させているのに対し、本アルゴリズムでは直前の再送発生時の値まで急速に拡大するという方式を採用している。性能評価の結果、伝送遅延の大きなボトルネックリンクのあるネットワークにおいて、既存の TCP よりもスループットを向上させることを明らかにした。

参考文献

[1] Kevin Fall, et. al.,"*ns* Notes and Documentation," http://www.isi.edu/~kannan/nsDoc.ps.gz, Nov. 1997.