

インラインネットワーク計測に基づく TCP スループットの保証手法

山根木果奈[†] 長谷川 剛^{††} 村田 正幸[†]

[†] 大阪大学 大学院情報科学研究科 〒 565-0871 大阪府吹田市山田丘 1-5

^{††} 大阪大学 サイバーメディアセンター 〒 560-0043 大阪府豊中市待兼山町 1-32

E-mail: [†]{k-yamanegi, murata}@ist.osaka-u.ac.jp, ^{††}hasegawa@cmc.osaka-u.ac.jp

あらまし リアルタイム配信型アプリケーションなど、通信品質の確保を必要とするアプリケーションに対して、我々の研究グループではトランスポート層プロトコルである TCP を用いて一定のスループットを上位アプリケーションに提供する手法を提案している。これまでの研究では、シミュレーションによって提案手法の基本的な特性を明らかにしたが、制御に必要としているインラインネットワーク計測手法を用いた評価は行っていない。そこで本稿では、インラインネットワーク計測によって得られた利用可能帯域の情報に基づいた制御を導入し、より発展的なシミュレーションを行うことで、提案手法の本質的な性能を評価する。また本稿では、複数のコネクションの合計スループットを統合して管理する手法について、提案およびシミュレーション評価を行い、個々に制御を行う場合よりも高い確率で TCP スループットを獲得できることを示す。

キーワード TCP (Transmission Control Protocol), スループット保証, 輻輳制御方式, シミュレーション

Achieving predictable throughput of TCP based on inline network measurement

Kana YAMANEGI[†], Go HASEGAWA^{††}, and Masayuki MURATA[†]

[†] Graduate school of Information Science and Technology, 1-5 Yamadaoka, Suita, Osaka, 565-0871, Japan

^{††} Cybermedia Center, Osaka University, 1-32 Machikaneyama, Toyonaka, Osaka, 560-0043, Japan

E-mail: [†]{k-yamanegi, murata}@ist.osaka-u.ac.jp, ^{††}hasegawa@cmc.osaka-u.ac.jp

Abstract The demand of Internet users for diversified services has increased due to the rapid development of the Internet, and applications requiring QoS guarantee, such as real-time media delivery services, have become popular. Our research group has proposed transport-layer approaches to provide such high-level quality of network services. In the present paper, we propose a congestion control mechanism of TCP for achieving predictable throughput. It does not mean we can guarantee the throughput, while we can provide the throughput required by an upper-layer application at high probability when network congestion level is not so high. We herein present the evaluation results for the proposed mechanism obtained in simulation experiments and confirm that the proposed mechanism can assure a TCP throughput if the required bandwidth is not so high compared to the physical bandwidth, even when other ordinary TCP (e.g., TCP Reno) connections occupy the link.

Key words TCP (Transmission Control Protocol), throughput guarantee, congestion control mechanism, simulation

1. はじめに

インターネットにおけるサービスにおいて、VoIP やテレビ会議システムなどのリアルタイム配信型アプリケーションが近年めざましく発展し、ユーザ数が爆発的に増加している。しかしながらこれらのアプリケーションが円滑に動作するためには、一定のネットワーク品質が必要となり、高いネットワーク品質への要求が高まっていると言える。例えば音声や映像などのリッチコンテンツを効率よく配信するためには、一定のスループットを提供することが要求される。また実時間ストリーミング配信アプリケーションなどは、伝送遅延や遅延ジッタがその品質に大きく影響する。

そのようなネットワーク品質を実現する手法の一つとして、IntServ [1] や DiffServ [2] などの IP 層における技術が挙げられるが、これらの手法はフローが通過するすべてのルータに品質制御機能が実装されている必要があり、ネットワーク規模に対するスケーラビリティ、導入コストなどの面から実現は困難であると考えられる。

そのようなネットワーク品質を実現する手法の一つとして、IntServ [1] や DiffServ [2] などの IP 層における技術が挙げられるが、これらの手法はフローが通過するすべてのルータに品質制御機能が実装されている必要があり、ネットワーク規模に対するスケーラビリティ、導入コストなどの面から実現は困難であると考えられる。

一方、動画ストリーミングなどのアプリケーションでは、トランスポート層プロトコルとして User Datagram Protocol (UDP) を用い、転送速度などの制御はアプリケーションが行うことによって、安定したアプリケーション品質を確保する手法が存在する。しかしこれらの手法は、アプリケーションの特性によって制御手法を変更する必要があり、アプリケーションごとに制御を組み替える非効率性が問題として指摘される。また、異なるアプリケーションが発生させるトラフィック間の相互干渉や公平性などを考慮することができないため、複数のネットワークアプリケーションが同時に使用される状況下では、アプリケーションの品質が不確定的になる。

以上のように、IP 層あるいはアプリケーション層において通信品質を維持・向上させることには、さまざまな問題点が存在する。そこで我々の研究グループにおいては、トランスポート層プロトコルを用いることで様々な通信品質を実現することに着目している。その一つとして、一定のスループットを TCP の制御によって獲得する手法を提案している [3]。本来 TCP は、ラウンドトリップ時間 (Round Trip Time: RTT)、パケット廃棄率、および競合するフロー数などによって、そのスループットが大きく影響を受ける [4] ため、一定のスループットを 100% 確実に獲得することはできない。しかし、アプリケーション品質を維持するためにスループットを高い確率で獲得することができれば、アプリケーション品質の向上が期待できると考えられる。これまでの研究では、ネットワークの輻輳レベルに関係なく、高い確率で一定のスループットを獲得することのできる、TCP の輻輳制御方式を提案した。本提案手法では、TCP の輻輳制御方式のうち、ウィンドウサイズの増減アルゴリズムを変更し、送信側 TCP のみの変更によって、アプリケーションに必要とされるスループットを高い確率で獲得することを実現している。本手法を用いることによって、例えば、Windows Media Player [7] や Real One Player [8] など、TCP トラフィックをバッファリングすることによって動画ストリーミングを行うアプリケーションの品質を大幅に向上させることができると考えられる。

提案手法は、我々の研究グループで提案している Inline Measurement TCP (ImTCP) [5] によって計測された利用可能帯域に関する情報を利用することによって、ネットワークの輻輳状態を考慮した制御を行う。ImTCP が行うインラインネットワーク計測は、送信側ホストで設定したデータパケットの送信間隔に対して、その ACK パケットの到着間隔の変化を観察することによって利用可能帯域を計測する手法である。ImTCP は、TCP コネクションがデータ転送に用いるデータパケットと ACK パケットのみを用いてネットワークの利用可能帯域を計測するため、計測用パケットを必要としない。さらに、ImTCP の機構は TCP 層の最下層部分に実装され、輻輳制御などの TCP 本来の機能からは分離することができる [6] ため、TCP の他の機能に影響を与えることなく、利用可能帯域を計測する機能を TCP に追加することができるという特徴を持つ。したがって提案手法においても、ImTCP アルゴリズムを用いて利用可能帯域に関する情報を得ることができる。

これまでの研究においては、提案手法は最新かつ正確なネットワークの利用可能帯域値を得ることができると仮定してシミュレーションを行った。それによって提案手法の基本的な特質を明らかにし、提案手法はアプリケーションから指定されたスループットを高い確率で獲得できることを実証した。本稿で

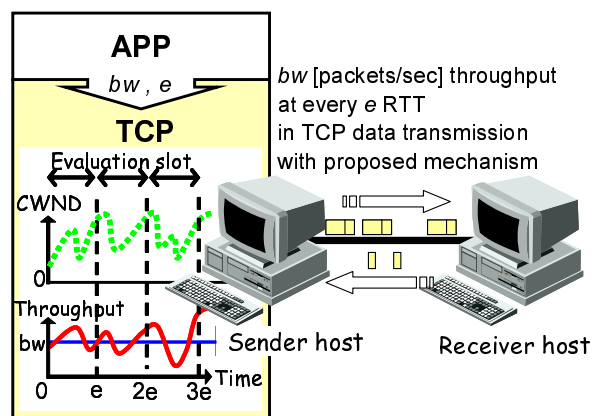


図1 提案手法の概要

は、提案手法に ImTCP による制御を導入してシミュレーションを行い、計測によって得られた利用可能帯域値に基づいた制御を行うことで提案手法の総合的な性質を明らかにする。

さらに本稿では、1 本のコネクションを制御する手法を拡張することにより、複数の TCP コネクションをまとめて管理・制御することによって、複数コネクションの合計スループットを確保する手法を提案する。複数コネクションをまとめてある一定のスループットを確保することができれば、例えば広域イーサネットサービスなどを用いて複数拠点ネットワークが相互接続されたような企業ネットワークにおいて、特定の拠点間のトラフィックに対して一定のスループットを提供するなどのサービスが可能となる。

提案手法は、ns-2 [9] を用いたシミュレーションを行うことで評価する。提案手法を用いるコネクションとして、動画ストリーミングアプリケーションを想定した無限長データの転送を行う long-lived コネクションだけでなく、Web トラフィックを想定した生存時間の短い short-lived コネクションが混在した複数のコネクションをまとめて制御する場合についても評価を行う。

本稿の構成は以下のとおりである。2 章では提案手法の説明を行い、3 章ではシミュレーションにより提案手法の性能を評価する。最後に 4 章でまとめと今後の課題を述べる。

2. 提案手法

2.1 1 本のコネクションを制御する手法

図 1 は、本提案手法の概要を図で示したものである。提案手法においては、上位アプリケーションから目標スループット (required throughput) および時間間隔が指定されることを想定している。ここで目標スループットを bw (packets/sec)、時間間隔を e (RTT) とすると、提案手法は e RTT ごとの平均スループットが目標スループット bw 以上になることを目的としている。指定された時間間隔 e は評価スロット (evaluation slot) と呼ばれ、さらに評価スロットを複数の制御スロット (control slot) に分割する (図 2)。つまり制御スロット長を s とすると、 $2 \leq s \leq e$ である。提案手法は、制御スロットごとにスループットの目標値 (target throughput) g を設定し、その値に基づいて輻輳ウィンドウサイズの増加量を動的に変動させることによってデータ転送速度を制御する。評価スロットよりも細かな時間間隔である制御スロットをサイクルとした制御を行うことによって、評価スロット終了時にその期間の平均スループットが bw 以上になることを意図している。

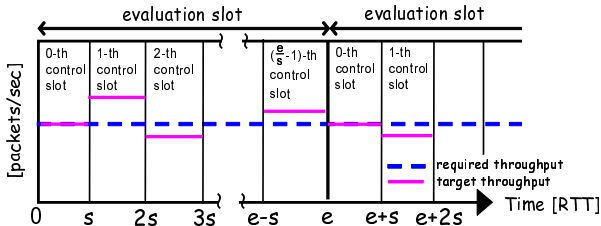


図2 評価スロットと制御スロット

提案手法は、ACK パケットが到着するたびに次式の k に適切な値を代入することによって輻輳ウィンドウサイズの値を決定する。

$$cwnd \leftarrow cwnd + \frac{k}{cwnd} \quad (1)$$

k に代入する値は次のように決定する。 i 番目の制御スロット (以降、制御スロット i と呼ぶ) のスループットの目標値を g_i と表す。また制御スロット i が開始してから j 番目の ACK パケットが返ってきたとき、制御スロット i の平均スループットが g_i となるための輻輳ウィンドウサイズの増加量を k_j^{bw} と表す。 k_j^{bw} は、スループットの目標値の獲得に必要なパケット数を制御スロット内で全て送信できるように以下の式を用いて計算する。

$$k_j^{bw} = \frac{2\{g_i \cdot srtt_i \cdot s - a_j - (s - n_j - 1)cwnd_{n_j}\}}{(s - n_j - 1)(s - n_j)}$$

ここで $srtt_i$ は、制御スロット i が開始したときに TCP が持つ RTT の指数移動平均である smoothed RTT (sRTT) の値であり、また a_j は、現在の制御スロットが開始してから j 番目の ACK パケットが返ってくるまでに送信したパケット数である。また j 番目の ACK パケットが返ってきたとき、現在の制御スロットが開始してから n_j RTT 経過しているとおくと、 $cwnd_{n_j}$ は、現在の制御スロットが開始してから n_j RTT 経過したときの輻輳ウィンドウサイズの値を示す。

提案手法は、(1) 式における k の値に関して、下限値 k_{min} および上限値 k_{max} を設定する。これはネットワークが十分空いている場合にも、提案手法が TCP Reno と同等のスループットを獲得することを可能する一方で、ネットワークが混雑している場合に、輻輳ウィンドウサイズの増加量が極度に大きく設定されることによる性能低下を防ぐためである。すなわち輻輳ウィンドウサイズの下限値 k_{min} は、TCP Reno が輻輳ウィンドウサイズの増加量として採用している 1 に設定し、また上限値 k_{max} は、ImTCP アルゴリズムによって計測された利用可能帯域の値に基づき、ネットワークの空き帯域を考慮した値に設定される。

以上をまとめると、(1) 式の k は k_j^{bw} 、 k_{min} および k_{max} を考慮して次の条件で決定される。

$$\begin{cases} \text{if } \{(k_j^{bw} < k_{min}) \text{ or } (k_{max} < k_{min})\} & k \leftarrow k_{min} \\ \text{elseif } \{k_j^{bw} > k_{max}\} & k \leftarrow k_{max} \\ \text{else} & k \leftarrow k_j^{bw} \end{cases}$$

提案手法の詳細は関しては [3] を参照されたい。

2.2 複数本のコネクションをまとめて制御する手法

本節では、前節で述べた 1 本のコネクションを制御する手法を拡張することによって、複数本のコネクションをまとめて管理し、その合計スループットが目標スループット以上となるこ

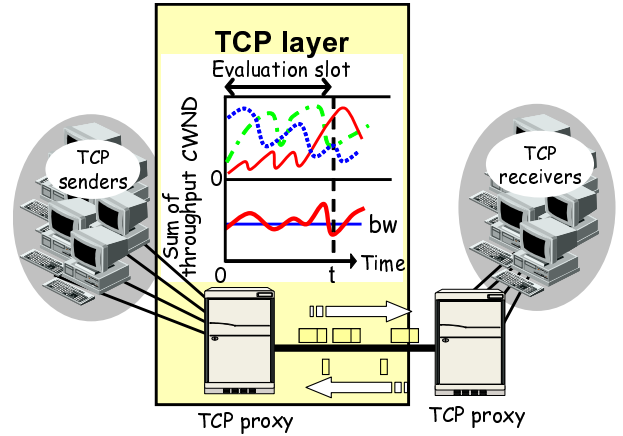


図3 複数本のコネクションを制御する手法

とを目的とした輻輳制御手法を提案する。この手法は、TCP プロキシ [10] などのトランスポート層プロキシにおいて用いることを想定し、送受信ホストはプロキシに接続されていることを考える (図 3)。このとき各コネクションのスループットは送信ホスト側のプロキシにおいて制御され、管理する合計スループットはプロキシ間のスループットとする。

指定された目標スループットおよび時間間隔をそれぞれ bw (packets/sec)、 t (sec) とすると、制御の対象とする複数のコネクションにおいて、 t 秒間におけるプロキシ間の合計スループットが bw 以上になるように輻輳ウィンドウサイズの値を制御する。ここで送信ホスト側のプロキシが特定した、アクティブ状態にある TCP のコネクション数を N_{pm} とおく。このとき N_{pm} に含まれるコネクションは、輻輳回避フェーズにあるアクティブなコネクションとし、スロースタートフェーズにあるコネクションは制御の対象としない。これは、スロースタートフェーズはコネクション確立時やタイムアウト後にネットワーク帯域を探るフェーズであることから、スループットを獲得するために用いるべきではないと考えるためである。

評価スロットおよび制御スロットは、制御の対象となる複数のコネクションで共通の時間間隔を用いる。評価スロット長 e (RTT) は、送信ホスト側のプロキシが持つ sRTT の最小値 $srtt_{min}$ を用いて $e = t/srtt_{min}$ とし、また制御スロット長は s (RTT) とする。ここで評価スロット長および制御スロット長は RTT を単位とした変数であるが、提案手法は制御スロット i が開始した時点の sRTT の値を $srtt_i$ (sec) を用いて、制御スロット i のスロット長を $(srtt_i \cdot s)$ 秒と決定する。スロットの終了判定はタイマを用いず、以下のように行う。送信ホスト側のプロキシが制御の対象としているコネクションの ACK パケットを受け取った時、制御スロット i が開始してから $(srtt_i \times s)$ 秒以上経過していれば制御スロット i を終了し、制御スロット $i+1$ を開始する。

制御スロット i において、制御スロットのスループットの目標値を g_i とすると、 j 番目の ACK パケットを受け取ったとき、 $(srtt_i \cdot s)$ 秒間の平均スループットが g_i となるための輻輳ウィンドウサイズの増加量 k_j^{bw} は以下のようにして得る。

$$k_j^{bw} = \frac{2\{(g_i \cdot srtt_i \cdot s - a_j^{sum})/N_{pm} - (s - n_j - 1)cwnd_i^{n_j}\}}{(s - n_j - 1)(s - n_j)} \quad (2)$$

ここで a_j^{sum} は、現在の制御スロットが開始してから j 番目の ACK パケットを受け取るまでに制御の対象となるコネクショ

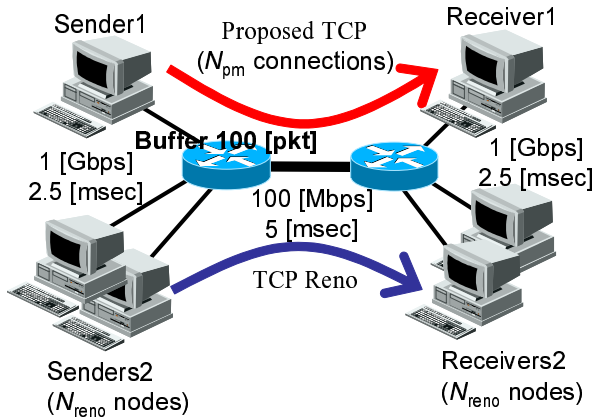


図4 シミュレーションモデル

表1 SURGE モデルのパラメータ

要素	生成関数	パラメータ
Size-Body	$p(x) = \frac{e^{-(\ln x - \mu)^2 / 2\sigma^2}}{x\sigma\sqrt{2x}}$	$\mu = 9.375; \sigma = 1.318$
Size-Tail	$p(x) = \alpha k^\alpha x^{-\alpha+1}$	$k = 133K; \alpha = 1.1$
Interval	$p(x) = \alpha k^\alpha x^{-\alpha+1}$	$k = 1; \alpha = 1.5$

ンが送信したパケット数の合計値である。この式によって、スループットの目標値 g_i を獲得するために必要なパケット数を、制御の対象となる N_{pm} 本のコネクションで分配してパケットを送信するように輻輳ウィンドウサイズの増加量を設定する。(2) は制御の対象となる全てのコネクションに対して用い、提案手法は、ACK パケットを受け取るたびに (2) 式を用いて k_j^{bw} を計算し、 k_{max} および k_{min} の値を考慮して 2.1 章と同様に輻輳ウィンドウサイズの値を設定する。

3. シミュレーションによる性能評価

本章では、提案手法の性能を ns-2 を用いたシミュレーションによって評価する。シミュレーションに用いるネットワークモデルを図 4 に示す。ボトルネックリンクの帯域および伝播遅延時間は 100 Mbps (12500 packets/sec に相当) および 5 msec とする。ボトルネックルータは DropTail ルータを使用し、バッファサイズは 100 packets とする。提案手法を用いる送受信ホストはそれぞれ 1 台存在し、その間の TCP コネクション数を N_{pm} と表す。また背景トラフィックを発生させる TCP Reno の送受信ホストはそれぞれ N_{reno} 台存在し、アクセスリンクの帯域および伝播遅延時間はそれぞれ 1 Gbps および 2.5 msec とする。パケットサイズは 1000 Bytes とする。提案手法のパラメータとして、 $e = 32$ とする。なおこの環境においては、32 RTT は約 1 秒に相当する。制御スロット長 s の初期値は e の 1/2 である 16 とする。

また本シミュレーションでは、無限長ファイルの転送を想定した long-lived フローに加えて、Web トラフィックを想定した short-lived フローを用いるが、その際には Scalable URL Reference Generator (SURGE) モデル [11] に基づき TCP コネクションのデータサイズやデータ転送間隔を決定する。SURGE モデルは、Web サーバにアクセスするユーザの挙動をモデル化し、実際の Web ワークロードを生成することができる。表 1 に、本シミュレーションで用いた SURGE モデルのパラメータを示す。

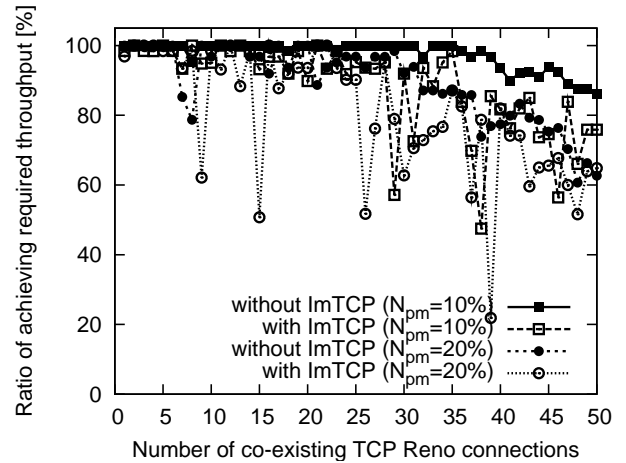


図5 Percentage of evaluation slot required throughput achieved

3.1 ImTCP の導入による影響

本節では ImTCP によるインラインネットワーク計測手法を用いることにより、利用可能帯域の情報に基づいた制御を導入することが、提案手法の性能に与える影響を評価する。ここでは $N_{pm} = 1$ とし、1 本のコネクションを管理する場合について評価する。提案手法のパラメータとして目標スループットを $bw = 1,250$ および $2,500$ (packets/sec) と設定する。これはボトルネックリンクの帯域の 10% および 20% にそれぞれ相当する。図 5 は、背景トラフィックを生成する TCP Reno コネクション数に対する、指定されたスループットを獲得できた評価スロット数の割合を示したものである。ImTCP を用いた制御を導入した場合のシミュレーション結果および、ImTCP を用いずに送信ホストが最新かつ正確な利用可能帯域を取得可能と仮定した場合のシミュレーション結果を示す。ここでのシミュレーション時間は 60 秒である。

図 5 より、ImTCP を用いない場合のシミュレーション結果と ImTCP を用いた場合のシミュレーション結果を比較すると、ImTCP を用いない場合の方がより安定した確率でアプリケーションから要求されたスループットを獲得できていることが分かる。それは背景トラフィックの本数が多い場合により顕著に表れている。これは背景トラフィックが少なくネットワーク帯域が十分余裕がある場合は、提案手法が利用可能帯域の情報に基づいた制御をほとんど行わないためである。またスループットを獲得できる確率が不安定になる原因としては、背景トラフィックが多い場合に、ネットワーク中に送出されたパケットがボトルネックルータにバッファリングされるため、パケットの到着時間間隔を用いて利用可能帯域を計測する ImTCP の計測結果に誤差が生じることが考えられる。しかしネットワークの空き帯域がほとんどない場合であっても、高い確率で要求されたスループットを獲得できていることは、提案手法による制御が効果的に機能しているためである。

次に、背景トラフィックである TCP Reno の輻輳ウィンドウサイズに最大値を設定した場合の評価を行う。これはアクセスリンクの帯域がボトルネックリンクの帯域より小さい環境を想定したものである。ここでは背景トラフィックの輻輳ウィンドウサイズの最大値を 100 packets と設定する。これは背景トラフィックである TCP Reno コネクションのスループットを 1 本あたり約 4Mbps に制限した場合に相当する。図 6 は、 bw をボトルネッ

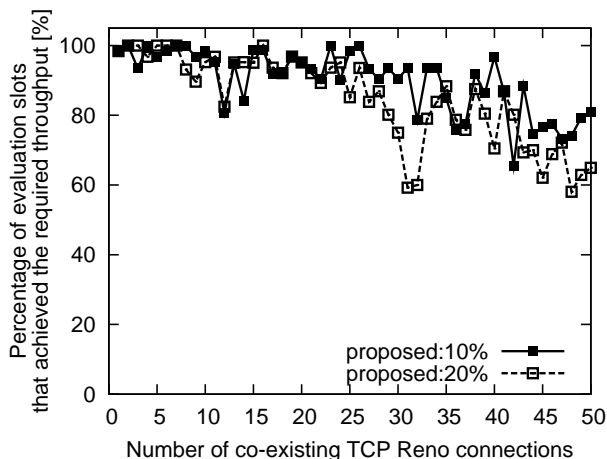


図6 競合する TCP Reno コネクションのスループットが制限される場合

クリンク帯域の 10% および 20% に設定した場合のシミュレーション結果である。ここでは $N_{pm} = 1$ とし、シミュレーション時間は 60 秒である。

図 6 より、図 5 で示された ImTCP を用いる場合の結果と比較して、提案手法がより高かつ安定した確率で要求されたスループットを獲得できていることが分かる。これは背景トラフィックである TCP Reno コネクションが、輻輳ウィンドウサイズの最大値を設定されることでスループットが制限され、提案手法がより効率良くネットワーク帯域を利用することができるためであると考えられる。これよりアクセスリンクの帯域がボトルネックリンクの帯域より小さい環境においては、提案手法がより効果的に動作すると考えられる。

3.2 複数のコネクションをまとめて保証する手法の評価

本節では、2.2 節で提案した複数のコネクションをまとめて管理する手法について評価を行う。ここでは図 4 中の Sender 1 と Receiver 1 の間に複数のコネクションを確立し、提案手法は Sender 1 において複数のコネクションのスループットを管理する。また、送信ホストがネットワークの物理帯域および現在の利用可能帯域を取得可能であると仮定し、輻輳ウィンドウサイズの増加量の上限値 k_{max} の値を適切に設定できるものとする。

3.2.1 提案手法および背景トラフィックに long-lived コネクションを用いる場合

まず、提案手法および背景トラフィックに、無限長のデータ転送を行う long-lived コネクションを用いた場合の評価を行う。提案手法を用いるコネクション数 N_{pm} は 5 および 10 とし、 $bw = 2,500$ (packets/sec) とする。また背景トラフィックである TCP Reno コネクションの輻輳ウィンドウサイズの最大値を 100 packets と設定する。図 7 は、全評価スロット数に対して提案手法が指定されたスループットを獲得できた評価スロット数の割合を示している。図中の “Reno” は、提案手法の代わりに 10 本の TCP Reno コネクションを用いた場合の結果を示し、また “For one” は、 $bw = 250$ (packets/sec) と設定した 1 本のコネクションを制御する手法を 10 本同時に用いた場合の結果を示している。

図 7 より、TCP Reno を用いた場合は、背景トラフィックのコネクション数が 30 本以上になると要求されたスループットを獲得できないことが分かる。これは TCP Reno が競合する全てのコネクションとネットワーク帯域を共有することから、競合

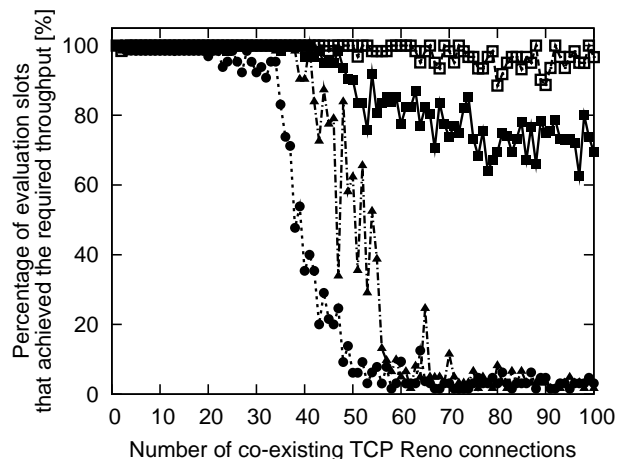


図7 複数のコネクションを用いる場合の性能

するコネクション数が増加するとスループットを獲得できないためである。また 10 本のコネクションを個々に制御する提案手法を用いた場合は、背景トラフィックのコネクション数が 40 本を超えるとその性能が急激に低下する。これは各コネクションが個別に利用可能帯域を計測し、その値に基づいたパケット数を複数のコネクションが同時に送信することによって、パースト的なパケットロス生じ、性能を低下させていることが原因として考えられる。一方複数のコネクションを管理する手法を用いた場合は、背景トラフィックのコネクション数が増加しても高い確率で要求されたスループットを獲得できていることが分かる。これは複数のコネクションで利用可能帯域に関する情報を共有することによって、同時に多量のパケットがネットワークに送出されることがないためである。さらに $N_{pm} = 10$ とした場合は、 $N_{pm} = 5$ とした場合よりも高い確率で目標スループットを獲得できることが分かる。これは、より多くのコネクションで分配してパケットを送信することで、より安定した性能を獲得できることを示している。

3.2.2 背景トラフィックに short-lived コネクションを用いる場合

次に、背景トラフィックに Web トラフィックを想定した生存時間の短い short-lived コネクションを用いた場合に、提案手法に与える影響について評価する。ここでは $bw = 2,500$ (packets/sec) とし、また $N_{pm} = 10$ とする。図 8 は、背景トラフィックである short-lived コネクションの平均スループットに対し、long-lived コネクションを用いた提案手法が要求されたスループットを獲得できたスロット数の割合を示している。また比較対象として、Sender 1 が通常の TCP Reno コネクションを用いた場合の結果を示している。図 8 より、提案手法が TCP Reno よりも高い確率で目標スループットを獲得できていることが分かる。しかしながら、図 7 で得られた結果と比較すると、背景トラフィックの増加に対して提案手法の性能が急激に低下している。これは short-lived コネクションのスループットが long-lived コネクションのスループットに比べてパースト的に変動するため、提案手法が short-lived コネクションから帯域を奪うことが困難であることが原因として考えられる。

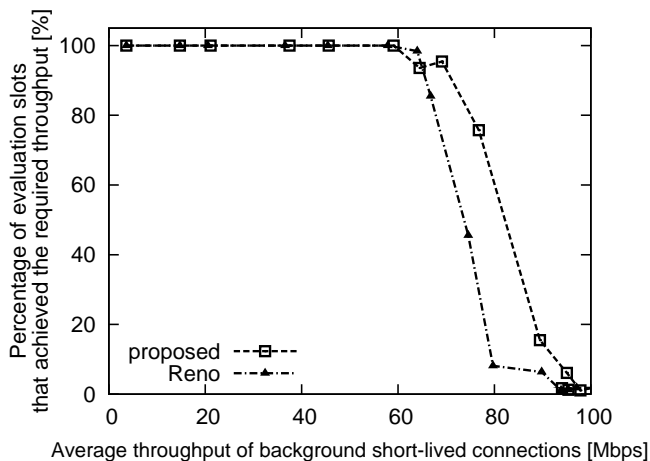


図8 背景トラフィックに short-lived コネクションを用いる場合

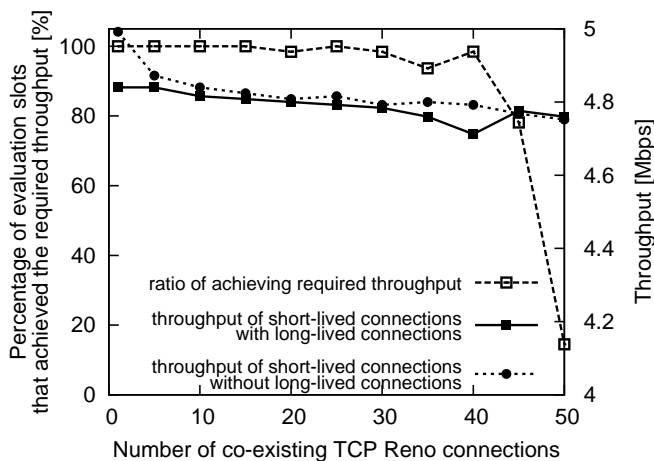


図9 提案手法が long-lived コネクションと short-lived コネクションを用いる場合

3.2.3 提案手法を用いる複数のコネクションに long-lived コネクションと short-lived コネクションが混在する場合

最後に、提案手法を用いる複数のコネクションに long-lived コネクションおよび short-lived コネクションが混在する場合について評価を行う。ここでは $bw = 2,500$ (packets/sec) とし、また $N_{pm} = 10$ としたうち 5 本のコネクションを long-lived コネクション、また残りの 5 本のコネクションを short-lived コネクションとする。図9は、提案手法が要求されたスループットを獲得できたスロット数の割合および short-lived コネクションの合計スループットの平均値を示している。またこの図は、提案手法を用いた long-lived コネクションが存在しない場合において、TCP Reno を用いた 5 本の short-lived コネクションの合計スループットの平均値も示している。図9より、提案手法は高い確率で要求されたスループットを獲得できていることが分かる。さらに short-lived コネクションの平均スループットは、提案手法を用いない場合とほぼ等しいと言える。これより、提案手法は long-lived コネクションを用いることによって要求されたスループットを獲得できると同時に、short-lived コネクションの性能に影響を与えないことが分かる。

4. おわりに

本稿では、一定のスループットを TCP の制御によってアプリケーションに対して提供することに着目し、複数の TCP コネクションの合計スループットについて、高い確率で一定のスループットを獲得することができる、TCP の輻輳制御方式を提案した。シミュレーションによる提案手法の評価結果から、インラインネットワーク計測手法を用いることによる提案手法への影響を明らかにした。また提案手法は背景トラフィック量が多い環境下であっても物理帯域の 20% に相当するスループットを高い確率で獲得できることが明らかとなった。

今後の課題としては、提案手法を実装し、実機を用いた実験によって提案手法の有効性を明らかにすることが挙げられる。

文 献

- [1] J. Wroclawski, "The use of RSVP with IETF integrated services," *RFC 2210*, Sept. 1997.
- [2] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated service," *RFC 2475*, Dec. 1998.
- [3] 山根木 果奈, 長谷川 剛, 村田 正幸, "スループット保証を実現する TCP の輻輳制御方式の提案と評価," 電子情報通信学会 技術研究報告 (IN2005-189), Mar. 2006.
- [4] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: a simple model and its empirical validation," in *Proceedings of ACM SIGCOMM'98*, Sept. 1998.
- [5] L. T. M. Cao, G. Hasegawa, and M. Murata, "An Inline measurement method for capacity of end-to-end network path," in *Proceedings of IM'2005 E2EMON Workshop 2005*, May 2005.
- [6] T. Tsugawa, G. Hasegawa, and M. Murata, "Implementation and evaluation of an inline network measurement algorithm and its application to TCP-based service," in *Proceedings of NOMS 2006 E2EMON Workshop 2006*, Apr. 2006.
- [7] Microsoft Corporation, "Microsoft Windows Media - Your Digital Entertainment Resource," available from <http://www.microsoft.com/windows/>.
- [8] RealNetworks Corporation, "Rhapsody & RealPlayer," available from <http://www.real.com/>.
- [9] T. V. Project, "UCB/LBNL/VINT network simulator - ns (version 2)," available from <http://www.isi.edu/nsnam/ns/>.
- [10] I. Maki, G. Hasegawa, M. Murata, and T. Murase, "Throughput analysis of TCP proxy mechanism," in *Proceedings of ATNAC 2004*, Dec. 2004.
- [11] P. Barford and M. Crovella, "Generating representative web workloads for network and server performance evaluation," *Measurement and Modeling of Computer Systems*, pp. 151-160, July 1998.