TCP congestion control mechanism for achieving predictable throughput

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

大阪大学　大学院情報科学研究科
情報ネットワーク学専攻　中野研究室
山根木 果奈

研究背景および目的

- 一定の通信品質を要求するアプリケーション
  - リッチコンテンツの配信、定常的に発生するデータの配信
  - 動画像のリアルタイム配信アプリケーション

⇒ネットワークに対する通信品質の確保が求められている

- 従来のアプローチ
  - IP層における制御
    ⇒スケーラビリティ、コスト面に問題
  - アプリケーション層における制御
    ⇒アプリケーションごとの制御組み換えによる非効率性
    ⇒異なるアプリケーション間の相互干渉

- トランスポート層において一定の通信品質を獲得する手法を提案する

利点
  - エンドホストの変更のみで実現可能
  - トラヒック間の干渉を考慮した制御を組み込むことが可能

欠点
  - 100%通信品質を保証することは不可能

TCPによって一定スループットの確保を目指す手法

- 一定のTCPスループットを確保する手法の提案

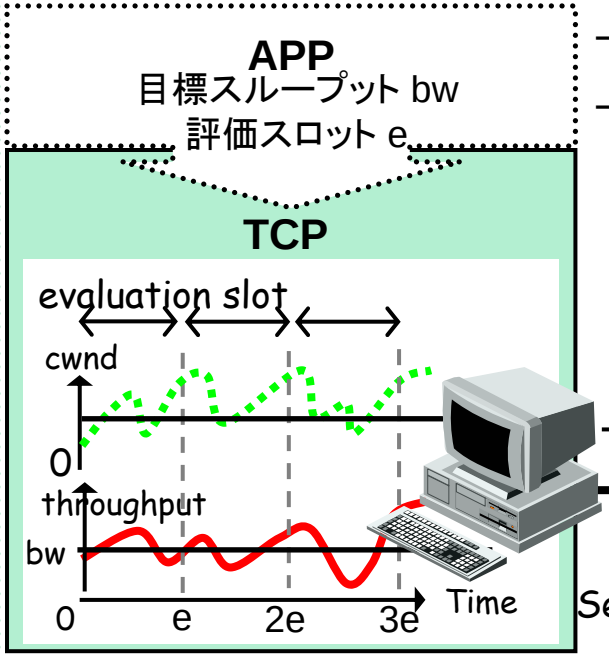
目的
- 輻輳時も一定のスループットの獲得する

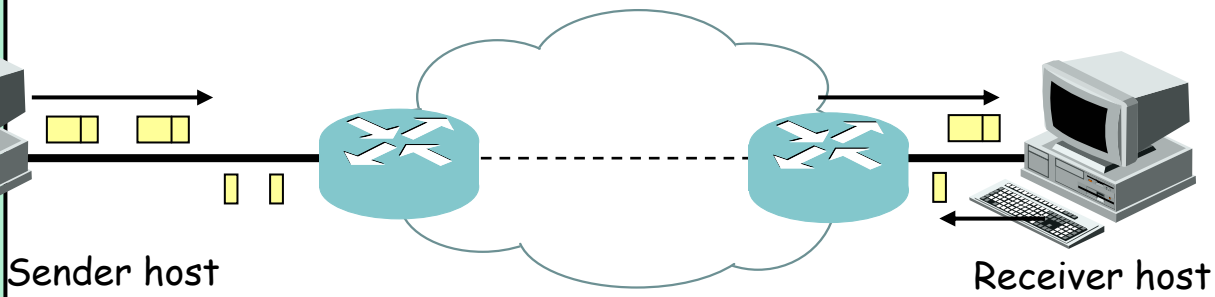
想定アプリケーション
- TCPストリーミングアプリケーション
- 定常的に発生するデータの転送 (観測データなど)

アプリケーションから与えられる
一定期間⇒評価スロット
一定値⇒目標スループット

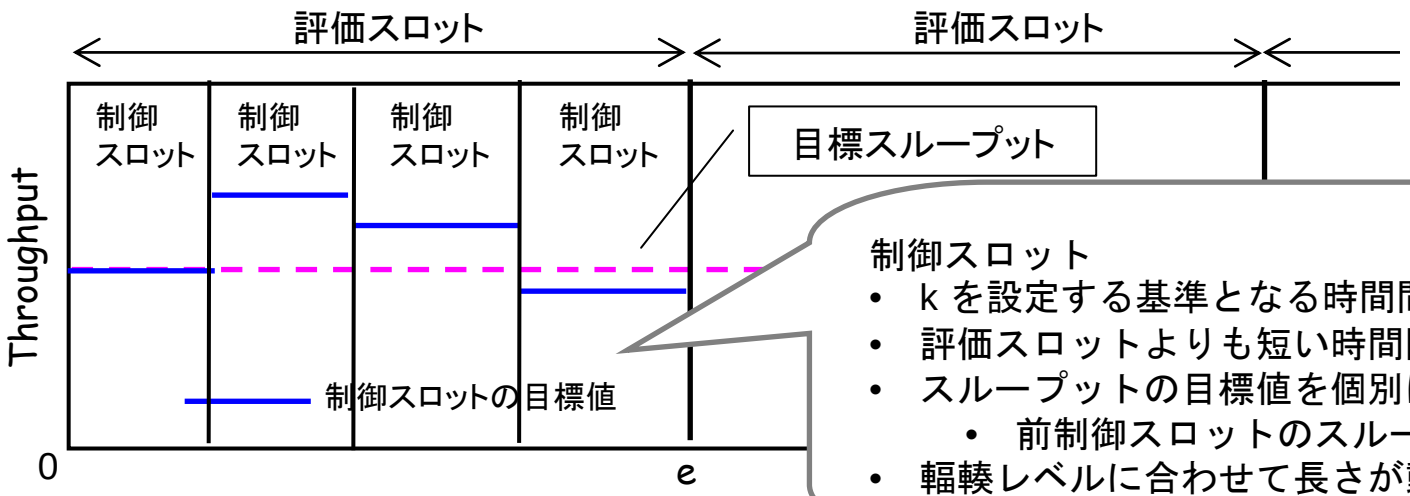
一定期間の平均スループットが一定値以上になるように制御
- 送信側TCPのみの変更
- 輻輳ウィンドウサイズ (cwnd) の増加量を動的に制御
- ロス時の cwnd の減少率はTCP Renoと同じ

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





輻輳ウィンドウサイズ増加量 k の設定



制御スロット
- k を設定する基準となる時間間隔
- 評価スロットよりも短い時間間隔
- スループットの目標値を個別に設定
  - 前制御スロットのスループット過不足を考慮
- 輻輳レベルに合わせて長さが動的に変動する

細かいサイクルを繰り返すことで、評価スロット終了時の平均スループットが目標スループットになることを意図している

目標値到達のために必要な送信パケット数を考慮し、
制御スロット終了までに必要なパケット数を全て送信できる増加量をkとして設定

$$k = \frac{2\{(g_i \cdot srtt_i \cdot s - a_j) - (s - n_j - 1)cwnd_{n_j}\}}{(s - n_j - 1)(s - n_j)} \quad (2)$$

gi （i番目の制御スロットの目標値）
$srtt_i$（i番目の制御スロットのSRTT）
s （制御スロット長）
aj（j番目のACKを受け取った時点のパケット送信数）
$cwnd_{nj}$（jのcwnd番目のACKを受け取った時点）

k の範囲の設定

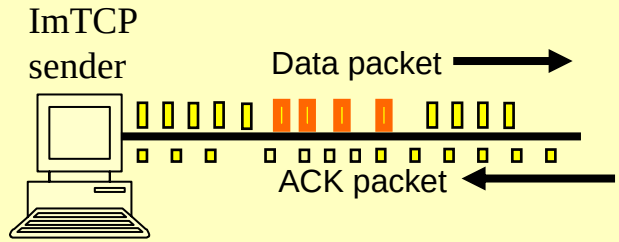
- k の範囲を設定する

$$k_{min} \leq k \leq k_{max}$$

  - k_{min}
    - k_{min}を設定しなければ、ネットワークが空いている場合も常に目標スループットを獲得
    - k_{min}=1(=TCP Reno)：ネットワークが空いている場合にTCP Renoと同等のスループットを獲得
  - k_{max}
    - 大きすぎるkが設定されることによるバースト的なパケットロスを防ぐ

ネットワークの空き帯域に応じて動的に設定する

- Inline measurement TCP (ImTCP) [11]
  - ネットワークの空き帯域を計測する手法
    - データ・ACKパケットのみを用いて計測を行う
  - ImTCP の特徴
    - 短い周期 (1 – 4 RTT) で継続的に計測を行うことができる
    - 少ない数のパケットで計測を行うことができる
  - ImTCP の問題点
    - 計測結果が不正確になる場合が存在する



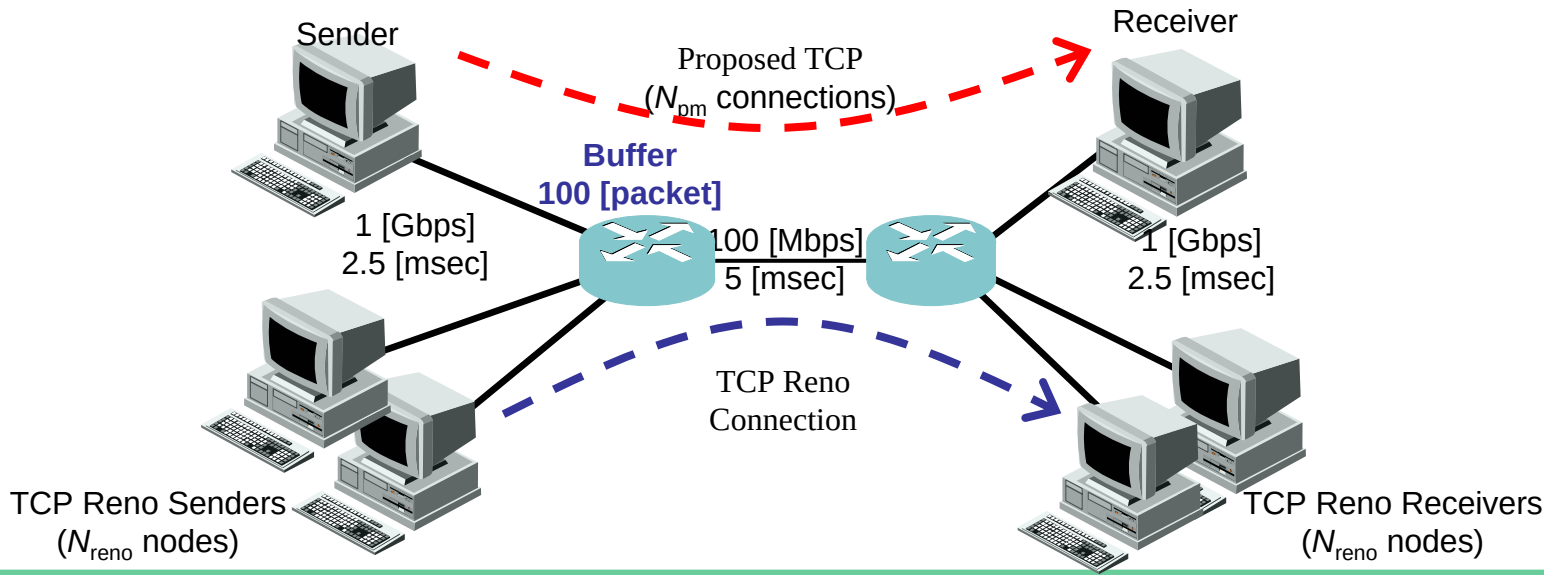
[11] C. L. T. Man, G. Hasegawa, and M. Murata, “ImTCP: TCP with an inline measurement mechanism for available bandwidth,” *Computer Communications Journal special issue of Monitoring and Measurements of IP Networks*, vol. 29, pp. 1614–1626, June 2006.

シミュレーションによる評価

- ns-2 を用いたシミュレーション

帯域	100 Mbps
RTT	20 msec
ルータ	Droptail
	バッファ 100 packets
パケットサイズ	1000 Bytes
クロストラヒック	TCP Reno

sender	提案手法の送信ホスト
receiver	提案手法の受信ホスト
TCP Reno Senders	背景トラヒックの送信ホスト
TCP Reno Receivers	背景トラヒックの受信ホスト



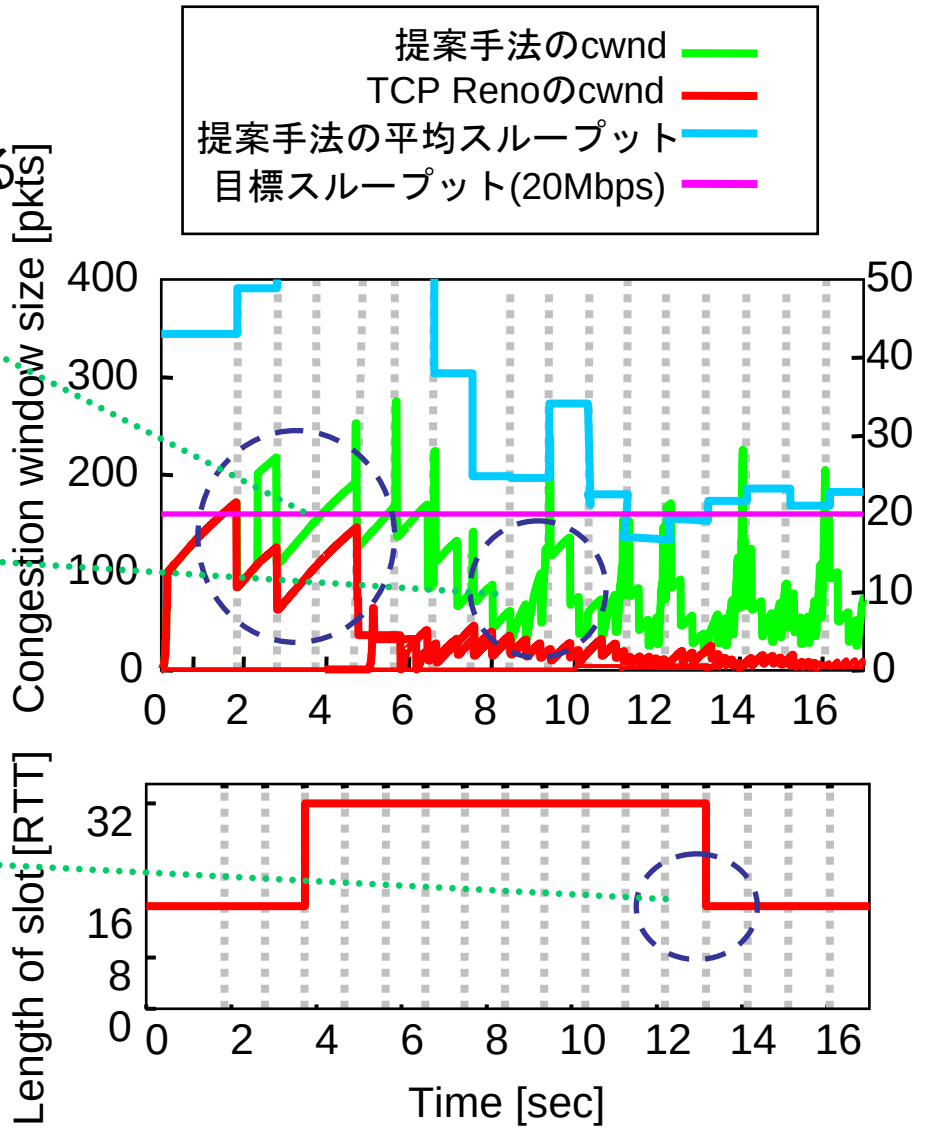
輻輳ウィンドウサイズとスループットの変動

- 提案手法を用いるコネクション 1本
  - 目標スループット 20Mbps
  - 評価スロット長 約1秒
- 背景トラヒック量を5秒ごとに変化させる
  - 1本→10本→40本

背景トラヒック1本
- ネットワーク帯域が十分空いている
- TCP Renoと同様の動作

背景トラヒック10本
- TCP Renoと同様の動作では目標スループットを確保できない
- 増加量kを大きくし、目標スループットを確保

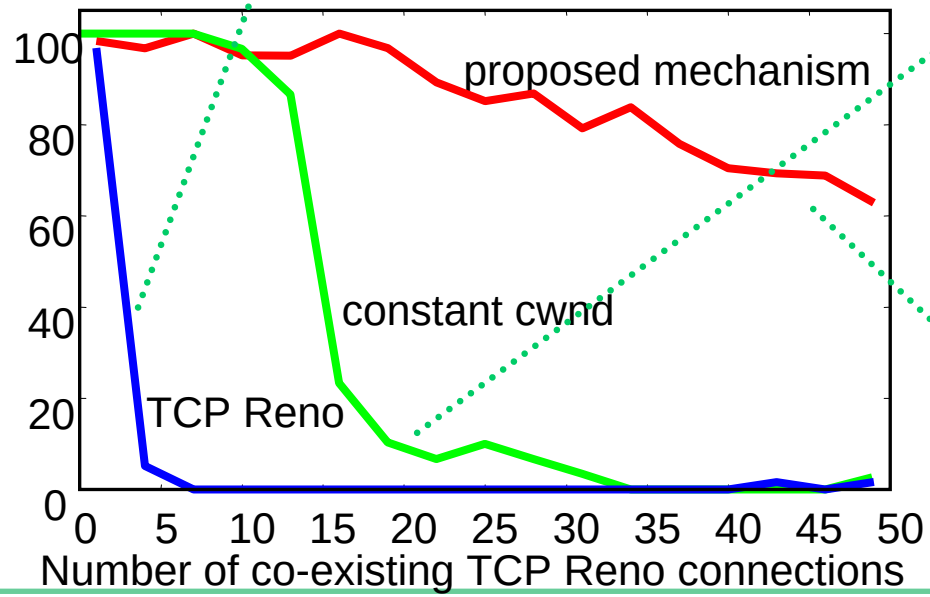
背景トラヒック40本
- k を大きくするだけでは目標スループットを獲得できない
- 制御スロット長を小さくして、制御を細かく



目標スループットを獲得できた割合

- 提案手法を用いるコネクション1本、目標スループット 20 Mbps
- シミュレーション時間 120秒
- 評価指標　目標スループットを獲得できたスロット数/全評価スロット数

用いたプロトコル		
提案手法 ———	TCP Reno ———	constant cwnd ——— (cwnd=srtt*bw 固定したTCP)



TCP Reno
- 背景トラヒックが4本以上になると20Mbpsを獲得できない

⇒ 全てのコネクションと平等にコネクションを分け合うため

constant cwnd
- 背景トラヒック量が多くなると目標スループットを獲得できない

⇒輻輳状態でも多量のパケット量を送信するため、バースト的なパケットロスが生じデータ転送効率が悪くなる

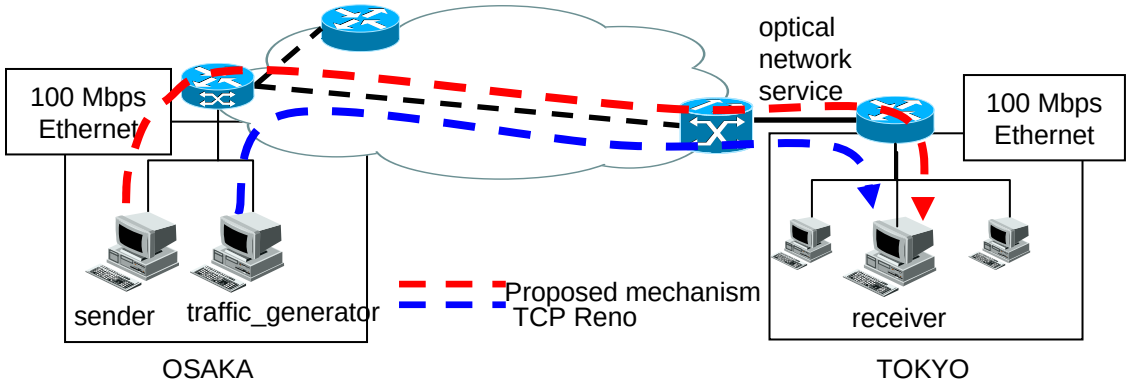
提案手法
- 背景トラヒックが増加しても高い確率で目標スループットを獲得

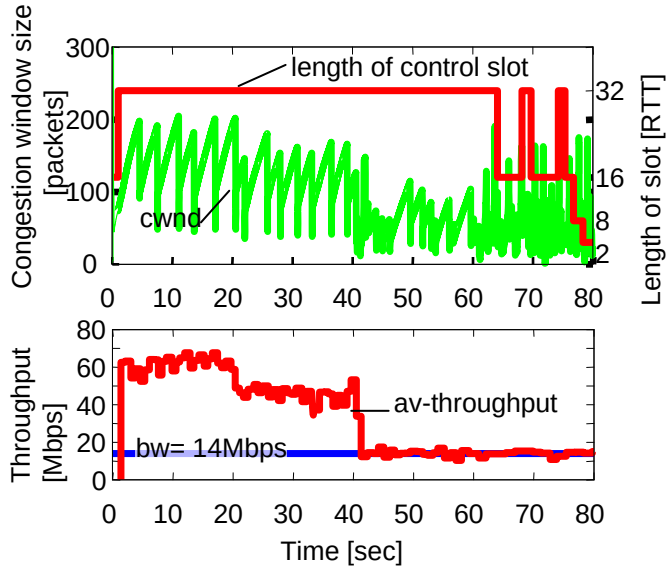
⇒輻輳レベルに応じた動的制御を行っているため

提案手法の実装

- 提案手法を実機に実装
  - Linux 2.6.16.21 カーネルシステム
  - 提案手法を輻輳制御モジュールとして実装
  - アプリケーションプログラムから目標スループット、評価スロット長を指定
    - setsockopt() システムコール
- 実網を用いた性能評価
  - 研究室内の実験ネットワーク環境
  - 東京-大阪間を結ぶインターネット環境

シミュレーションと同等の結果が得られた





- 実装コードを公開
  - http://www.anarg.jp/imtcp/

まとめ

- 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提供する輻輳制御手法を提案
  - TCPの輻輳ウィンドウサイズの増加量の変更のみで実現
  - ネットワークの空き帯域を考慮する
  - シミュレーションおよび実装実験において背景トラヒックが増加しても高い確率で目標スループットを獲得できることを実証

今後の課題

- 今回の実験とは異なるネットワーク環境を用いた性能評価
- 動画のストリーミングアプリケーションを用いた実証実験