

# インライン計測に基づく 高速TCP輻輳制御方式の提案

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  
井口 智仁

# 発表内容

- 研究背景

- TCP (Transmission Control Protocol) の輻輳制御機構
  - TCP Reno の輻輳制御方式の概要と問題点

- 研究内容

- 新しい TCP の輻輳制御方式の提案
  - 帯域情報を取得し, それを用いたアルゴリズムを使用

- シミュレーションによる性能評価

- まとめと今後の課題

# TCP の輻輳制御機構

- 主な目的

- 輻輳の発生を避けながらリンク帯域を使い切る
  - 複数コネクション間の公平性を実現できるように

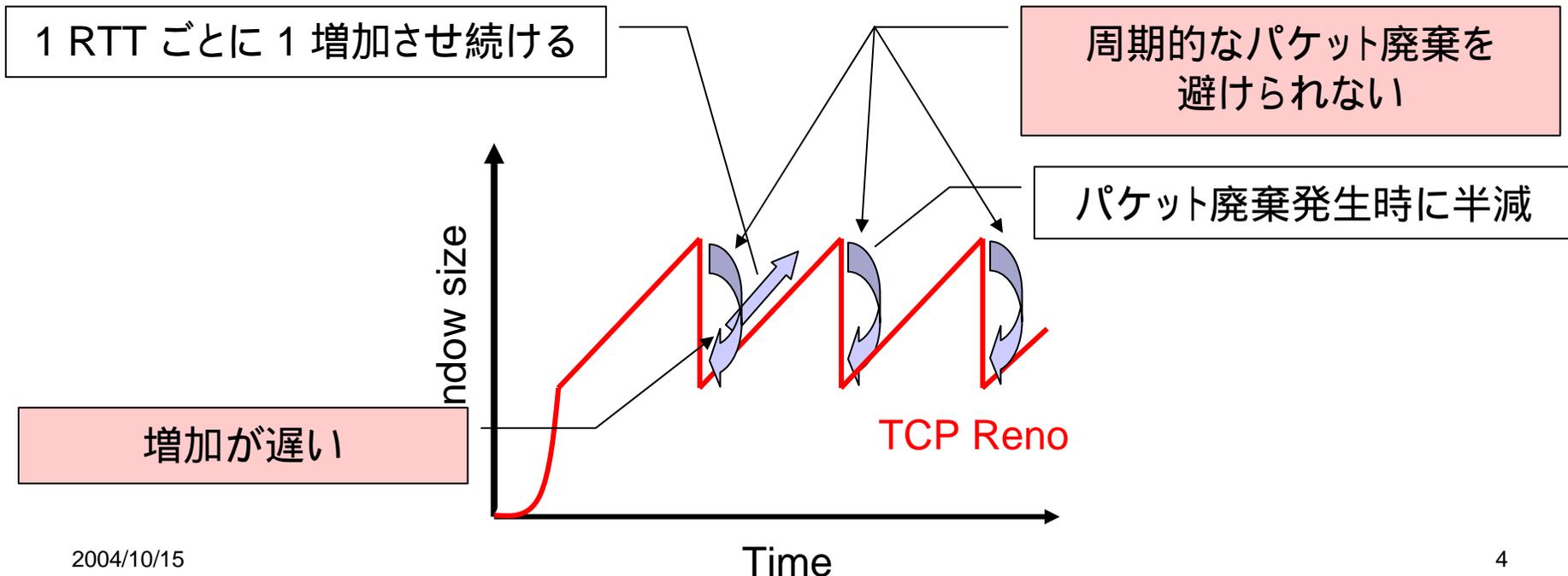
- ウィンドウサイズを増減させてデータの送出量を調節

- 帯域が空いているなら, 増加させる
- 輻輳のシグナルを受け取ると, 減少させる

ウィンドウサイズ：一度にネットワーク内へ送出可能なデータ量

# TCP Reno の輻輳制御方式の概要と問題点

- ウィンドウサイズの制御アルゴリズム
  - パケット廃棄の発生が輻輳発生シグナル
  - 帯域の情報を用いていない動作



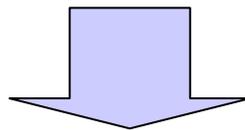
# 研究の目的

- **新しい TCP の輻輳制御方式の提案**
  - 何らかの方法を用いて、帯域の情報を取得する
  - 帯域情報を用いたウィンドウサイズ制御アルゴリズムを使用する

TCP Reno の問題点を本質的に解決する

# 帯域情報の取得

- ウィンドウサイズ制御アルゴリズムに必要な帯域情報を計測できる手法を使用
  - TCP の輻輳制御に直接利用可能な計測手法
    - 大量の計測用パケットを必要としない
    - 短い周期で継続的に結果を得られる



- インラインネットワーク計測手法
  - TCP コネクション内で帯域の計測を行う

# Inline Measurement TCP (ImTCP)

- インラインネットワーク計測手法のひとつ
- 特徴
  - 利用可能帯域, 物理帯域を計測可能
  - データパケットと ACK パケットのみを用いて計測
  - 短い周期 (1 ~ 4 RTT) で, 継続して計測結果を更新
  - TCP の性質を壊さないため, 任意の TCP に適用可能
- 計測方法の概要
  - データパケットの送信間隔を計測アルゴリズムで調節
  - 対応する ACK パケットの受信間隔から帯域を計算する

参考文献:

M. L. T. Cao, G. Hasegawa, and M. Murata, "Available bandwidth measurement via TCP connection," to be presented at IFIP/IEEE MMNS 2004, Oct. 2004.

# 輻輳制御の方針

- 帯域情報を使用したデータ転送速度の調節
  - データ転送開始時は、ゆっくりと増加させる
    - 複数コネクションの同時転送による輻輳を回避する
  - 帯域情報に応じて、増加速度を大きくする
    - いかなる大きさの帯域でも素早く使用する
  - 増加させ続けず、ある値に収束させる
    - パケット廃棄を引き起こさない

上記の動作を実現する、生物の個体数の変化を表す数学モデル

# ロジスティック増殖モデル

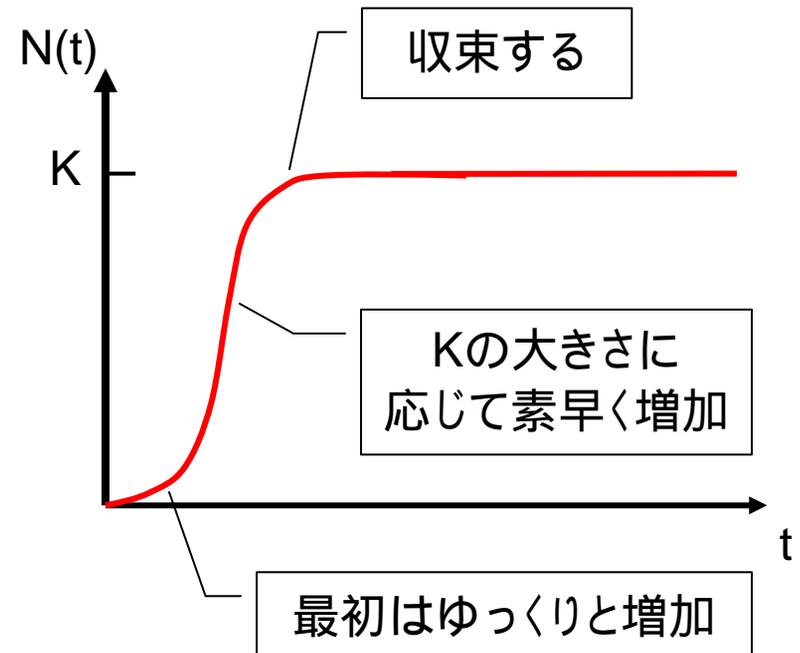
## ● 1 種の生物のある環境における個体数の変化

$$\frac{d}{dt} N(t) = \varepsilon \left( 1 - \frac{N(t)}{K} \right) N(t)$$

$N(t)$  : 個体数  
 $K$  : 環境容量  
: 内的自然増殖率 (  $< 2$  )

データ転送速度制御に適用

$N(t)$  : **データ転送速度**  
 $K$  : **物理帯域**



参考文献:

- ・J. D. Murray, *Mathematical Biology I: An Introduction*. Springer Verlag Published, 2002.
- ・寺本 英「数理生態学」朝倉書店, 1997

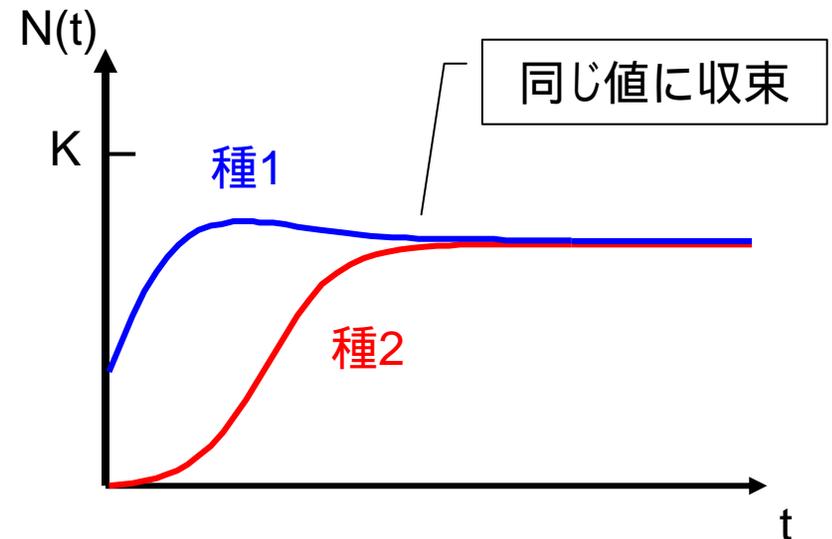
# ロトカ・ヴォルテラ競争モデル

- 2種の生物間で競争がある場合の個体数の変化
  - TCPコネクションが1本のみ存在, は現実的ではない

$$\frac{d}{dt} N_i(t) = \varepsilon_i \left( 1 - \frac{N_i(t) + \gamma_{ij} N_j(t)}{K_i} \right) N_i(t)$$

$ij$ : 競争相手種の存在による  
増殖率の低下率  
( $0 < \gamma_{ij} < 1$  が安定条件)

$i$ : 自分  
 $j$ : 相手



# 数学モデルの適用

- 現実的には, 3本以上のコネクションが存在
  - ロトカ・ヴォルテラ競争モデルを拡張する
- 他のコネクションの使用している帯域を直接知ることはできない
  - 物理帯域から利用可能帯域を引くことで近似する
- ウィンドウサイズ制御の式へ変換
  - ウィンドウサイズ      データ転送速度  $\times$   $base\_rtt$

$base\_rtt$ : 往復伝播遅延時間 (キューイング遅延を含まない)

# 提案方式

- 利用可能帯域  $A$ , 物理帯域  $K$  をインライン計測で取得する
  - 1 ~ 4 RTT ごとに更新
- 1 RTT ごとに, ウィンドウサイズ  $w$  の増加量を計算する
  - ACK パケット 1 つあたりの増加量 (1 RTT に  $w$  個の ACK)

$$\frac{d}{dack} w = \varepsilon \left( 1 - \frac{w + \gamma(K - A) \times base\_rtt}{K \times base\_rtt} \right)$$

- TCP Reno と同じ動作を行うケース
  - 帯域の計測結果が得られる前
  - パケット廃棄時
  - タイムアウト時

# 提案方式の特徴

- ウィンドウサイズの増加速度を帯域に応じて変更する
  - 帯域に対するスケーラビリティがある
- ウィンドウサイズを増加させ続けず、収束させる
  - 周期的なパケット廃棄の回避が可能
- 各コネクションのウィンドウサイズが同じ値に収束する
  - コネクション間の公平性がある

# シミュレーションによる性能評価

- シミュレーションについて

- ネットワークシミュレータ ns-2 を使用

- 式のパラメータは  $\alpha = 1.95$ ,  $\beta = 0.9$  とする

- 比較対象

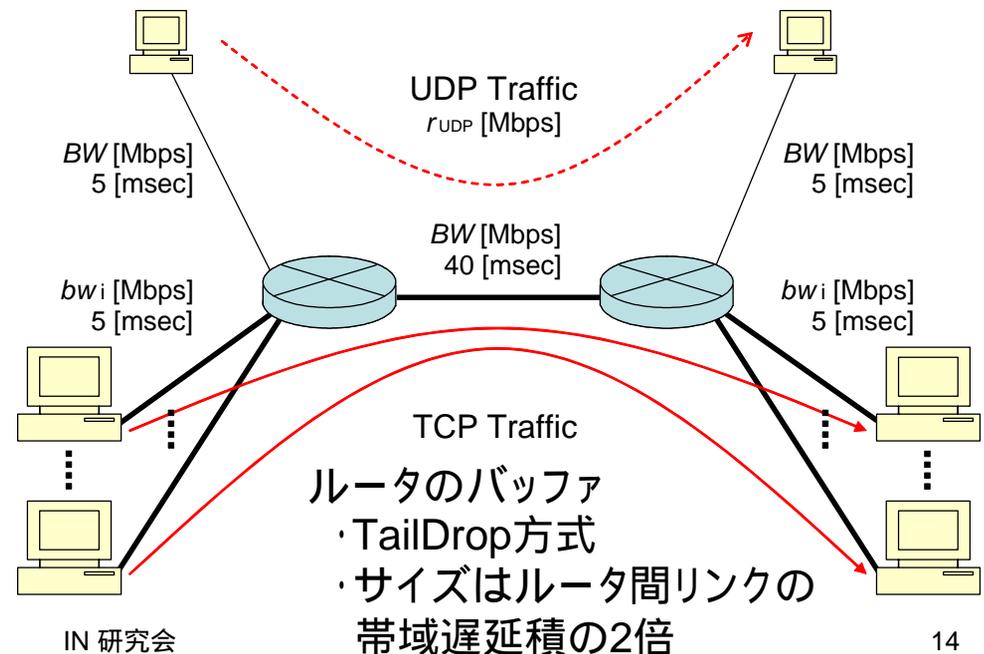
- TCP Reno

- HSTCP

- Scalable TCP

高速環境向けに  
改良された TCP

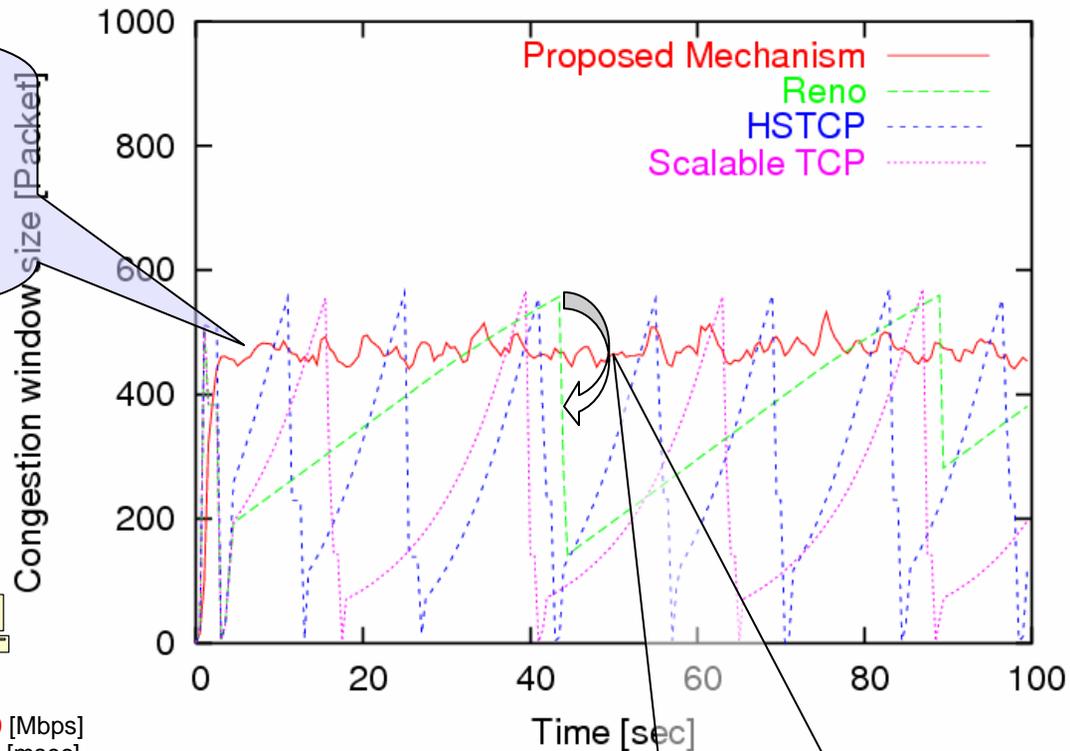
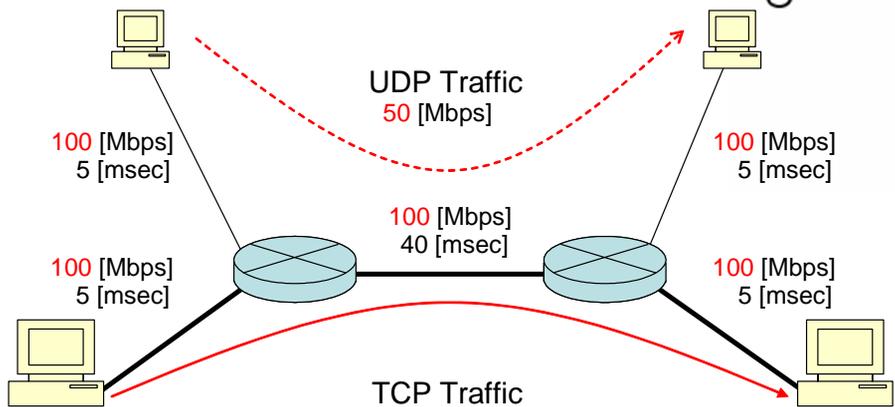
$N_{TCP}$



# 性能評価: 基本的な動作

提案方式は  
ウィンドウサイズが理想的な値に  
素早く収束し, パケット廃棄が  
発生しない

提案方式は  
リンク帯域を効率よく利用できる



Reno, HSTCP, ScalableTCP は  
周期的にパケット廃棄が発生する

# 性能評価：収束時間

BWが増加すると

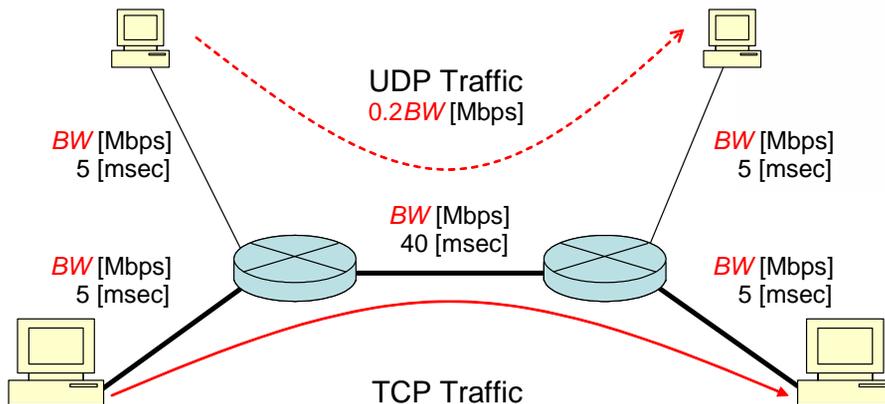
- ・HSTCP は収束時間が少しずつ大きくなる
- ・Scalable TCP は収束時間がほぼ同じ

収束時間の定義：

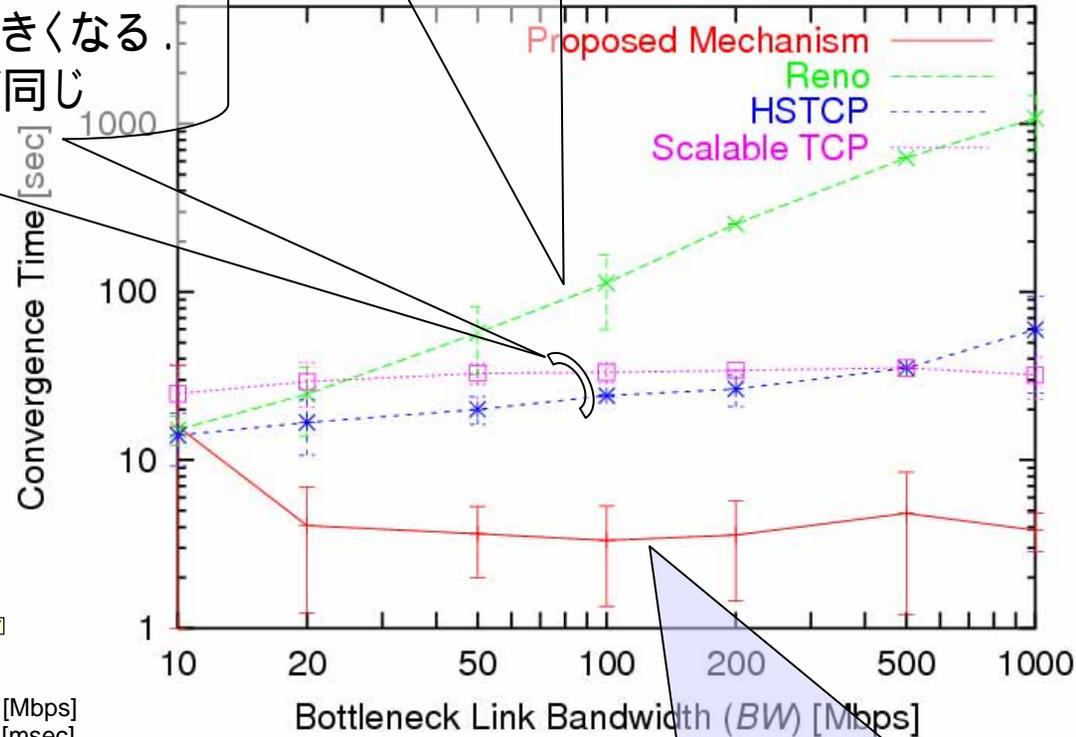
TCP コネクションがリンク帯域の99%を利用するまでに要する時間

リンク帯域 **BW**

= 10 Mbps ~ 1Gbps で変化



Reno は BWが増加すると収束時間が大きくなる

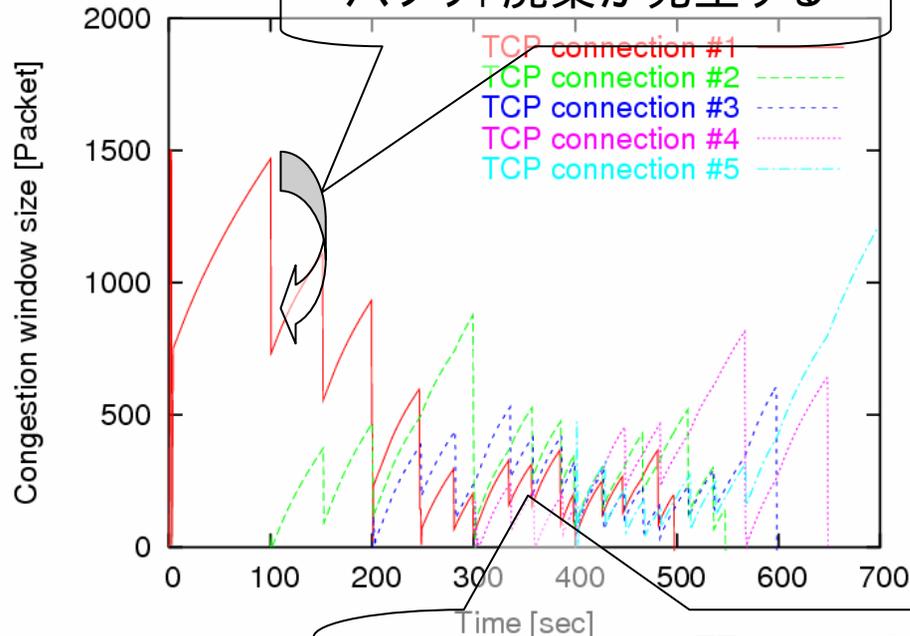


提案方式は

- ・BWが増加しても収束時間がほぼ同じ
- ・収束時間が小さい

# 性能評価: コネクション数の変化の影響

パケット廃棄が発生する

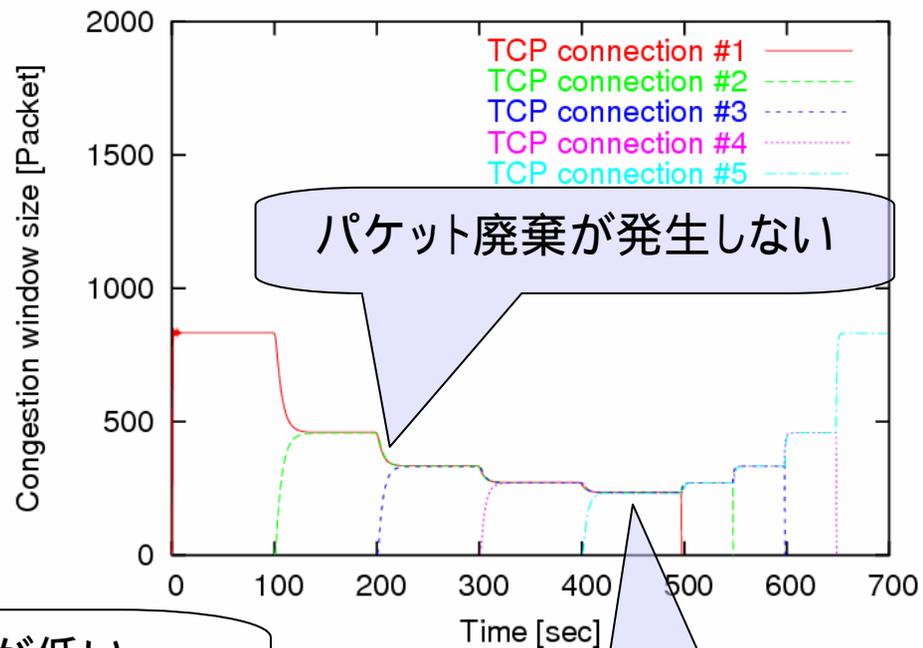


Reno

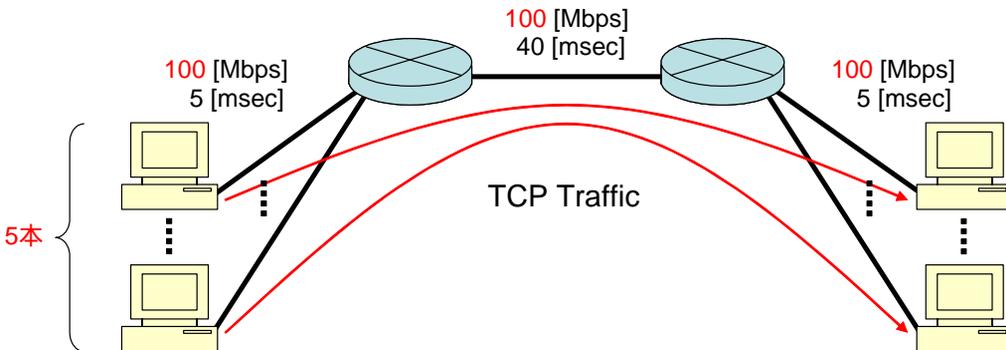
コネクション間の公平性が低い  
(公平な状態となるまで時間がかかる)

提案方式

パケット廃棄が発生しない



コネクション間の公平性がある  
(素早く公平な状態へ収束する)



# まとめと今後の課題

## ● まとめ

### ○ 新しい TCP の輻輳制御方式を提案した

- インラインネットワーク計測を用いて利用可能帯域, 物理帯域の情報を取得
- 数理生態学のモデルを適用したウィンドウサイズ制御アルゴリズムを使用

### ○ 現在の TCP よりも良い性能が得られることを示した

## ● 今後の課題

- 異なる RTT を持つコネクション間の公平性の実現
- TCP Reno と提案方式が混在する場合の評価
- 帯域の計測誤差による影響