A New Congestion Control Mechanism of TCP with Inline Network Measurement

Tomohito Iguchi
Osaka University
t-iguti@ist.osaka-u.ac.jp

Contents

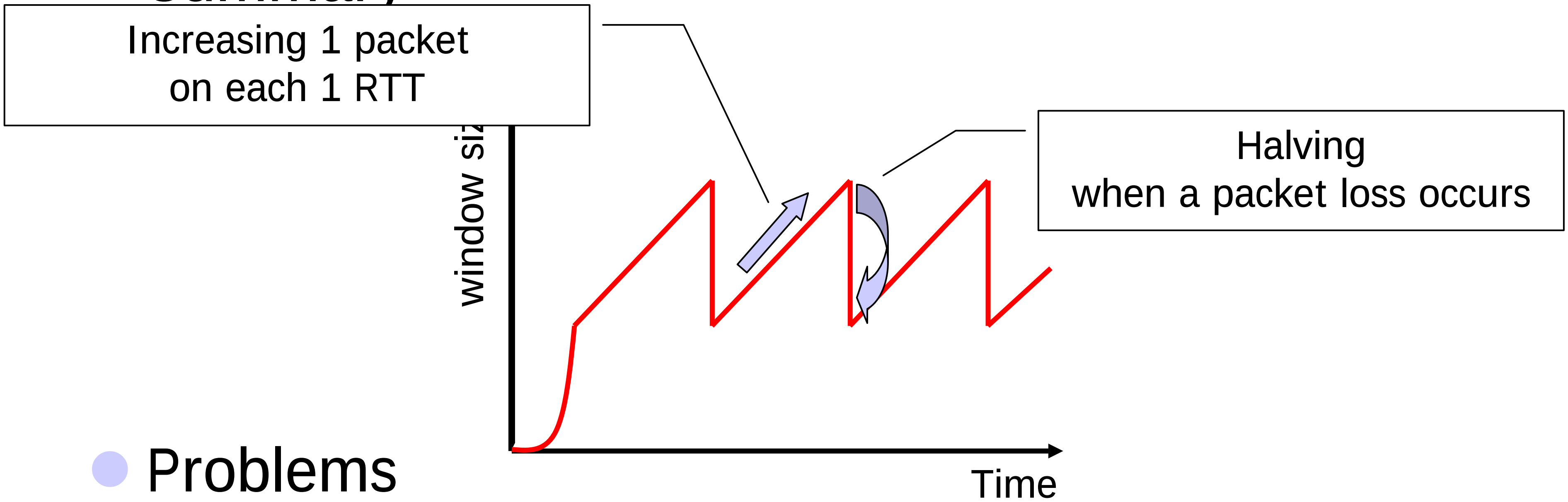
- Congestion Control Mechanism of TCP
  - TCP Reno's Window Size Control Algorithm
    - Problems and Reasons
- A New Congestion Control Mechanism
  - It obtains the bandwidth information by an inline measurement mechanism
  - It Controls the window size based on the bandwidth information
- Performance evaluation through simulations
- Conclusion and future work

Congestion Control Mechanism of TCP

- Main purposes
  - Avoiding network congestion and utilizing fully the link bandwidth
  - Realizing equal network bandwidth distribution among competing connections
- Basic idea
  - TCP adjusts data transmission rate by changing the window size.

TCP Reno's Window Size Control Algorithm

- Summary



- Problems
  - Low utilization of the link bandwidth in high-speed and long-delay networks
  - Periodic packet losses

Reasons and Solution of Reno's problems

- Reasons
  - The increase speed is fixed and small, and/or the amount of a decrease is too large
  - Reno doesn't recognize the bandwidth information
    - So it only increases window size
      until a packet loss occurs to fully utilize the link bandwidth
- Solution
  - Using the bandwidth information
    - The increase speed can be changed dynamically according to the bandwidth without packet losses

Objective of our study

- Proposing a novel congestion control mechanism of TCP to essentially solve the problem of TCP Reno
  - It requires the bandwidth information with the inline network measurement
  - It utilizes the bandwidth information to control its window size
    - The algorithm uses the mathematical model from biophysics

Inline Measurement TCP (ImTCP)

- Proposed in [1]
- Features
  - Using only data/ACK packets transmitted in TCP
    - No extra probe packets injected into the network
  - Yielding results every 1 〜 4 RTT
  - Obtaining the information of available and physical bandwidths

⇩

- The proposed mechanism uses this inline measurement mechanism
  - This mechanism can be directly employed into any TCP

Reference：
[1] M. L. T. Cao, G. Hasegawa, and M. Murata, “Available bandwidth measurement via TCP connection,” in *proceedings of IFIP/IEEE MMNS 2004*, Oct. 2004.

Design policy of congestion control mechanism

- Adjusting the data transmission rate using the bandwidth information
  - Increasing the rate according to the bandwidth
    - It has the scalability with link bandwidth
  - Converging the rate into a certain value
    - No packet losses occur

↓

- The proposed mechanism uses the mathematical model to describe changes in number of species

Lotka-Volterra competition model

~~predator-prey~~

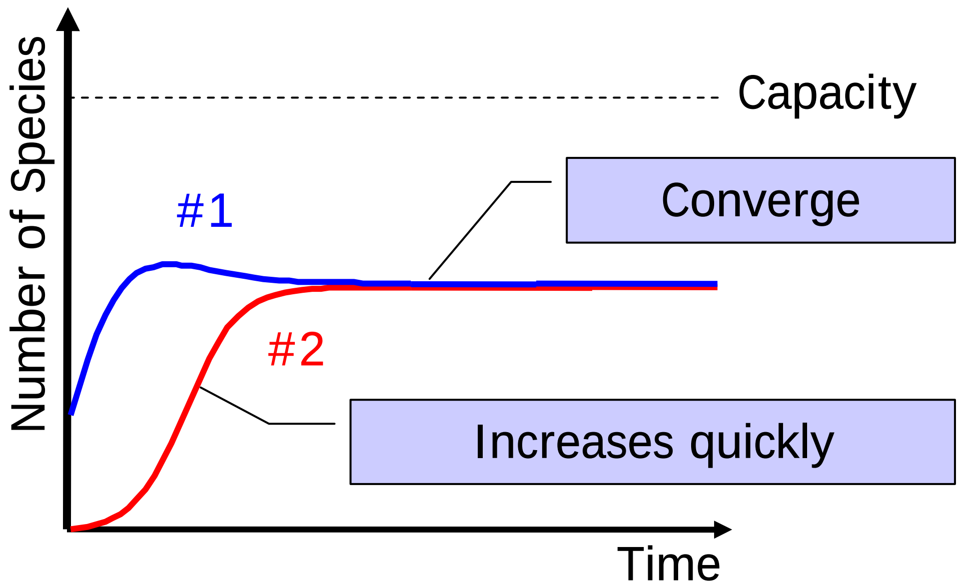
- Describing changes in number of 2 species
  - Famous model in biophysics [2]

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

$N_i(t)$: Number of species # i
K : Capacity
ε_i : Growth rate ($\varepsilon < 2$)
γ_{ij} : Competition coefficient
($0 < \gamma < 1$)

Apply to the data transmission control

$N_i(t)$: Data transmission rate
K : Physical bandwidth



Reference：
[2] J. D. Murray, *Mathematical Biology I: An Introduction*. Springer Verlag Published, 2002.

Converting the model into the congestion control algorithm

- Extend the equation for *n* connections
- Approximate the amount of the bandwidth used by the other connections
  - physical bandwidth – available bandwidth
- Change the equation
  → Using window size w_i $w_i = N_i \times RTT_{\min}$
- Convert the equation → per ACK
  - w_i ACK packets are received in 1 RTT

Proposed mechanism

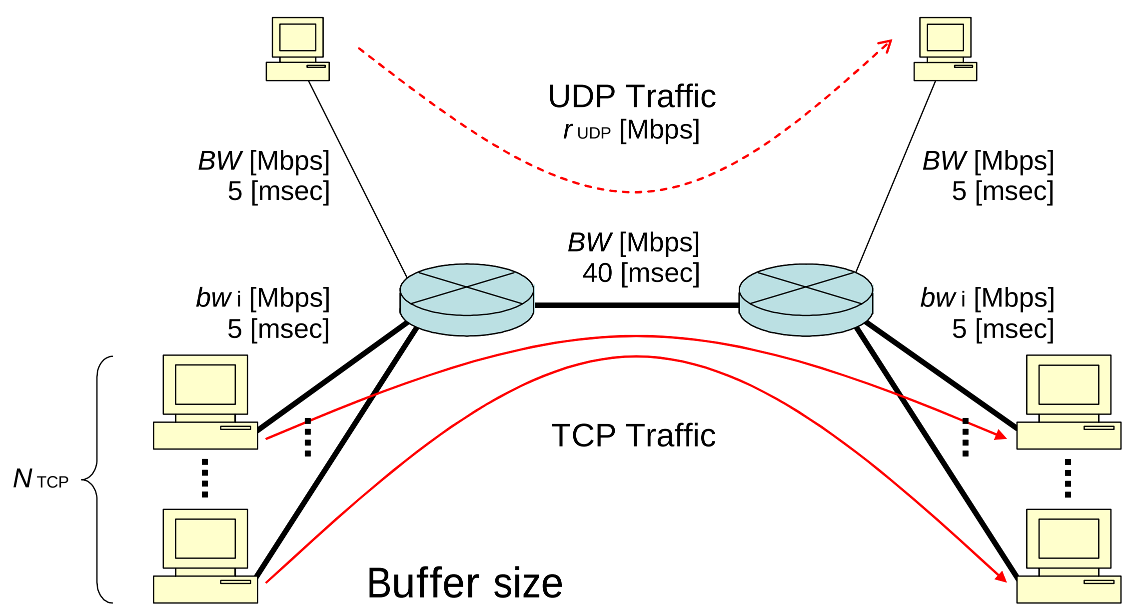
- Obtaining the bandwidth information from the inline measurement mechanism
  - Available bandwidth
  - Physical bandwidth
- Controlling the window size by using below equation:

$$\frac{d}{dAck}w_i = \varepsilon\left(1 - \frac{w_i + \gamma(K - A_i) \times RTT_{\min}}{K \times RTT_{\min}}\right)$$

Performance evaluation through simulations

- Details
  - The Network Simulator – ns-2
  - Set **ε** = 1.95, **γ** = 0.9
  - Comparison
    - TCP Reno
    - HSTCP
    - Scalable TCP

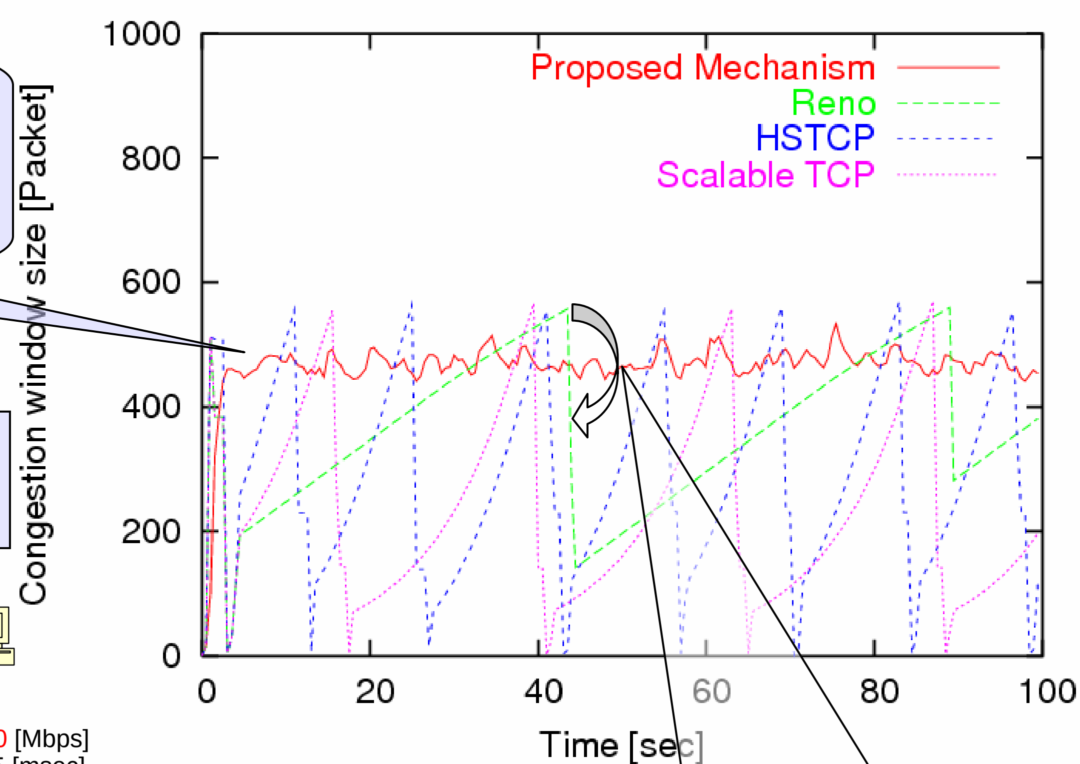
for high-speed and large bandwidth network



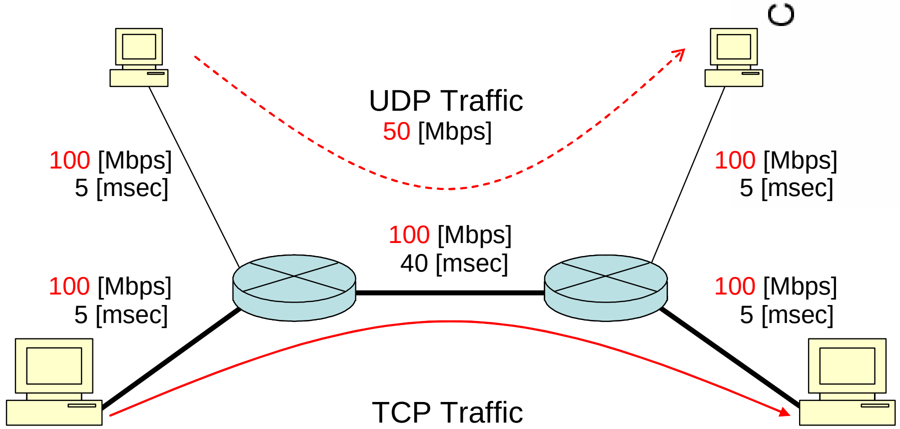
Buffer size
= Bandwidth-delay product

Fundamental behavior

Proposed mechanism:
- Convergence into ideal value
- No packet losses



Proposed mechanism can efficiently utilize the link bandwidth



Reno, HSTCP, Scalable TCP:
- periodic packet losses

Convergence Time

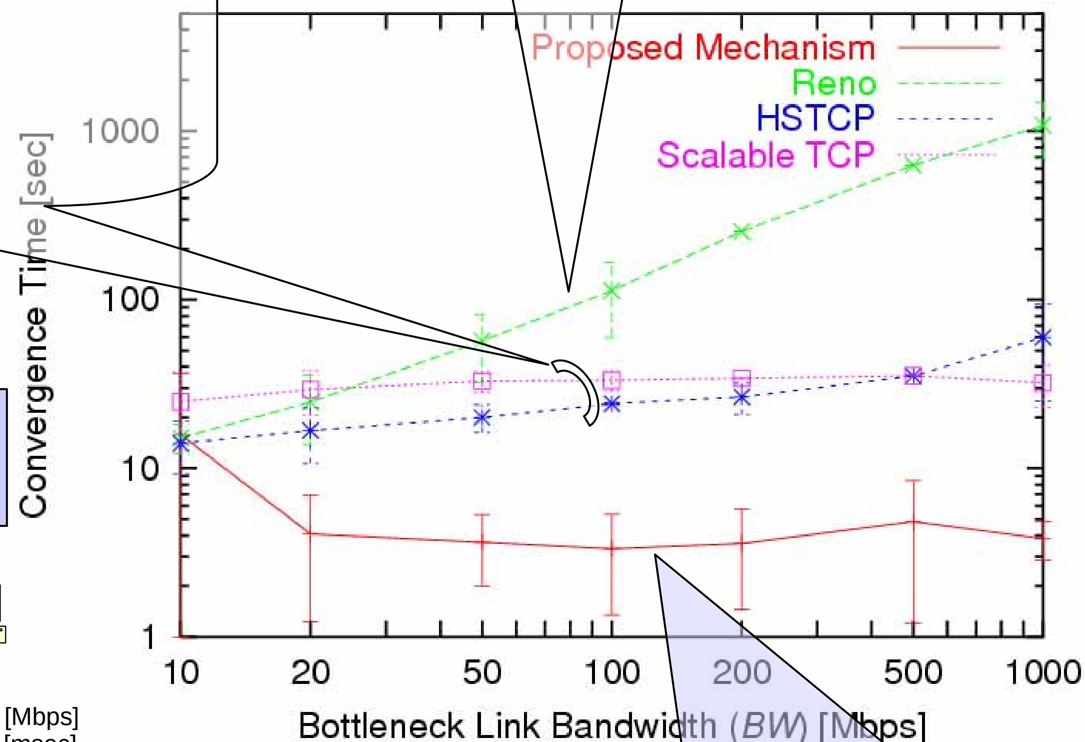
Reno:
Quite a large time

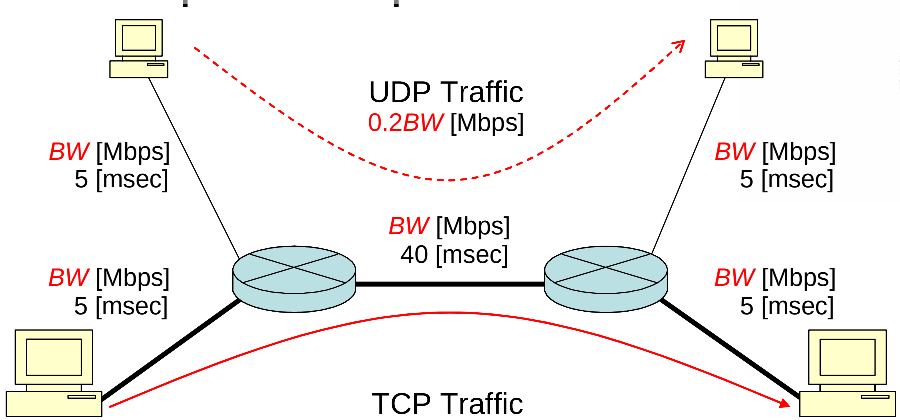
HSTCP:
The time that grows little by little
Scalable TCP:
Almost the constant time

Definition of convergence time：
the time it takes for the TCP connection

proposed mechanism has scalability with link bandwidth

= 10 Mbps ～ 1Gbps

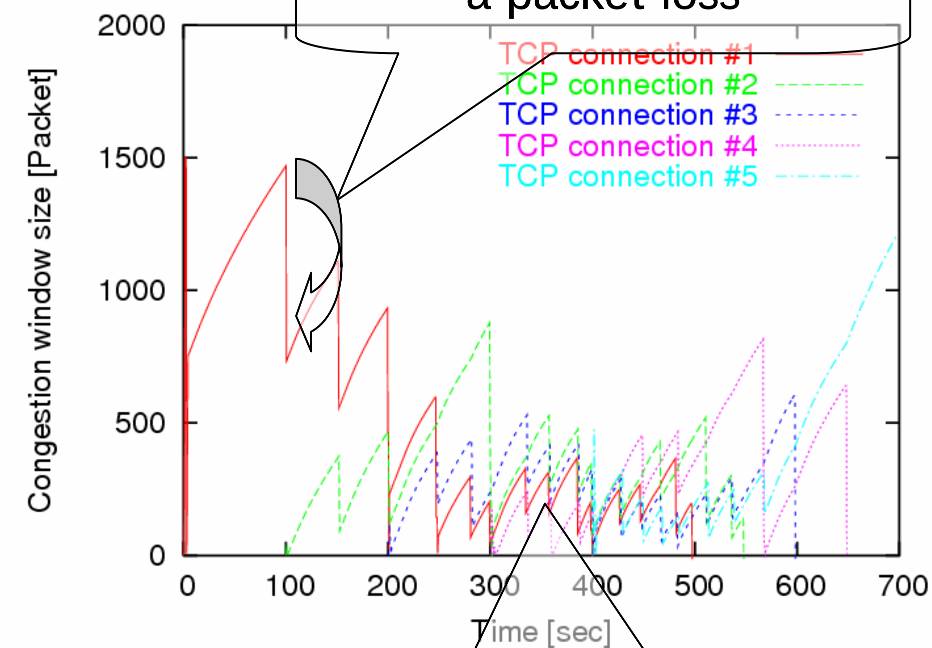




Proposed mechanism:
・Almost the constant time

Effect of changes in number of connections

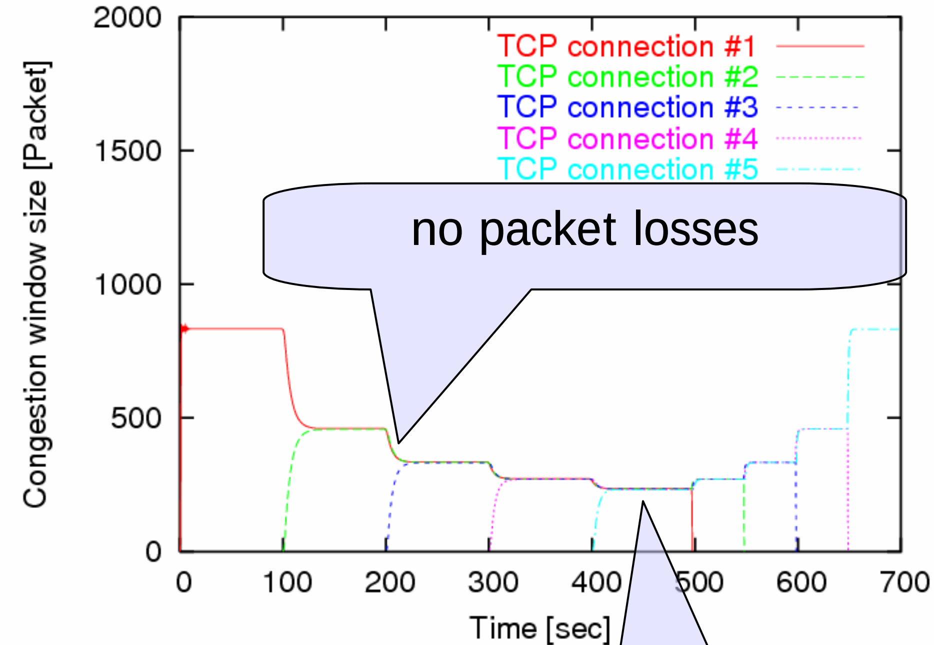
a packet loss



Reno

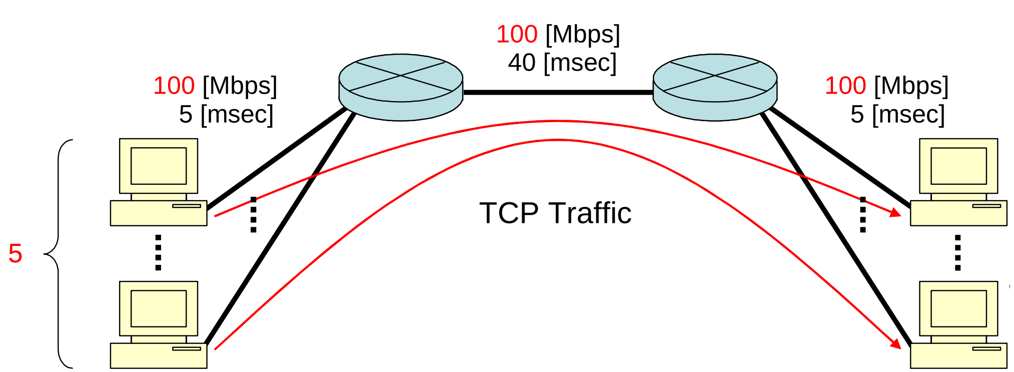
Unfair, Unstable

no packet losses



Proposed mechanism

Fair, Stable



Conclusion and Future Work

- We introduced a new congestion control mechanism of TCP
  - Features
    - It uses its bandwidth information obtained from inline measurements
    - It has the window size control algorithm based on the mathematical models from biophysics
  - Simulation results show that the proposed mechanism can improve the performance of TCP

- Future works
  - Fairness among connections with different RTTs
  - Fairness against TCP Reno connections
  - The effect of bandwidth measurement errors