Congestion control mechanism of TCP for achieving predictable throughput

----Osaka University----
Kana YAMANEGI, Go HASEGAWA, Masayuki MURATA

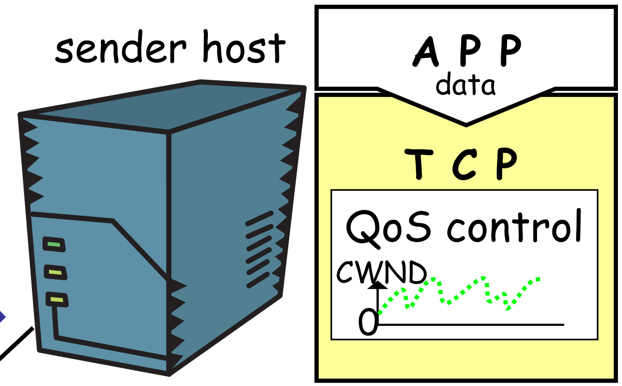
＊BACKGROUND＊

- delivery of rich contents
- video streaming application
- real time media delivery service

jitter
round trip time
throughput
...

require large and stable amount of network resources for maintaining Quality of Services (QoS)

sender host



＊EXISTING APPROACH＊

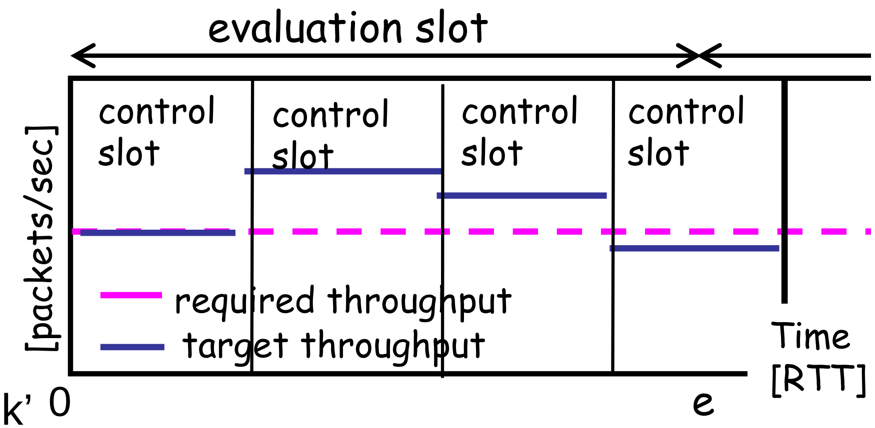
- IP layer approach (IntServ, DiffServ)
  - problems of scalability and costs
- application layer approach
  - cannot consider of fairness

We focus on achieving the predictable throughput by TCP connections

- achieve bw [packets/sec] throughput in e [RTT]
- control the congestion window size (cwnd)
  - change the degree of increase of the cwnd

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

＊PROPOSED MECHANISM (PM)＊

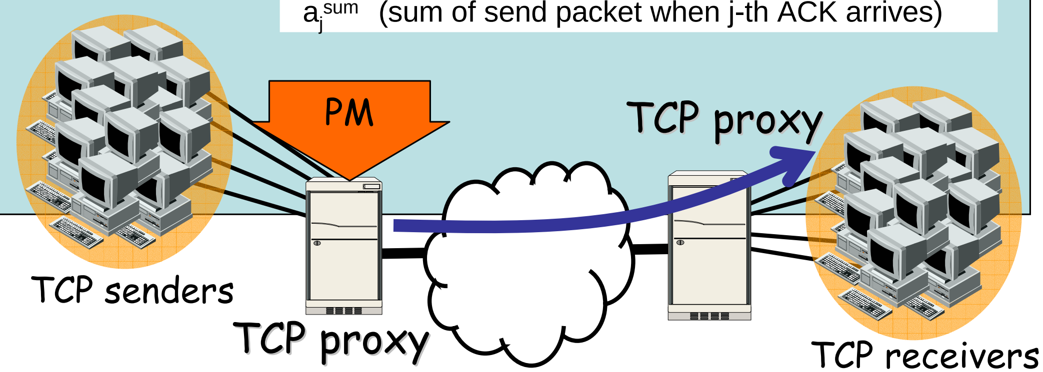


- evaluation slot
  - time interval required by application
  - average throughput in evaluation slot aims required throughput 'bw'
- control slot
  - time interval for setting the increase degree of cwnd 'k'
  - Each control slot sets the target throughput.
  - The length of control slot change according to the congestion level.

$$k = \frac{2\{(g_i \cdot srtt_i \cdot s - a_j^{sum}) / N_{pm} - (s - n_j - 1)\, cwnd_{n_j}\}}{(s - n_j - 1)(s - n_j)}$$

gi (target throughput of i-th control slot)
$srtt_i$ (sRTT of i-th control slot)
s (length of the control slot)
aj (number of send packets when j-th ACK arrives)
$cwnd_{nj}$ (cwnd when j-th ACK arrives)
a_j^{sum} (sum of send packet when j-th ACK arrives)

considering the number of send packets for achieving target throughput and the rest time of the control slot and the number of PM connections



extended to multiple connections

set the maximum and minimum value of k

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

k_{min} : 1 (=TCP Reno)

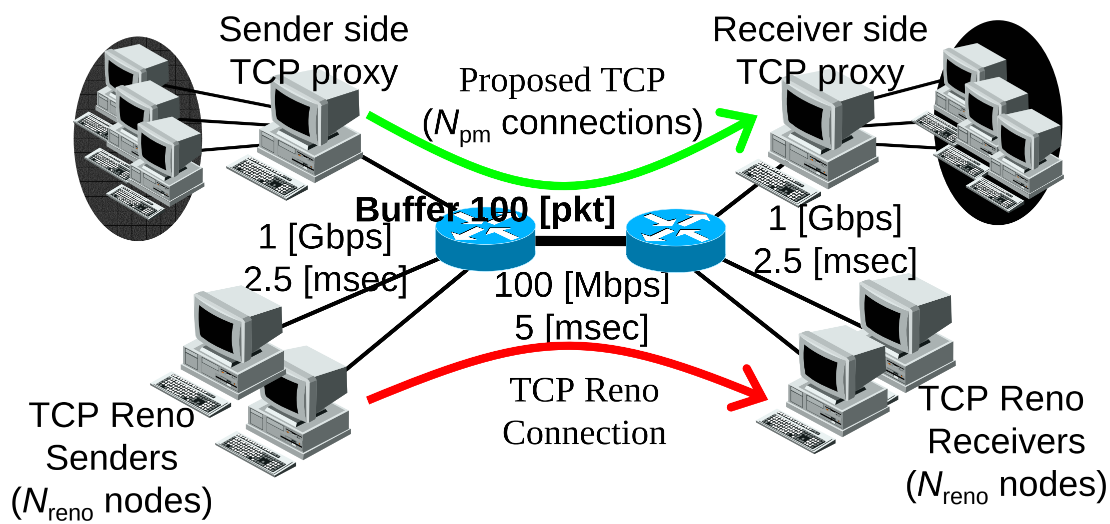
- get equivalent throughput to TCP Reno

k_{max}

- avoid bursty packet losses in the network

⇒set based on available bandwidth in the network

ImTCP
- measurement mechanism for available bandwidth
- use only data and ACK packets



＊NS-2 SIMULATION MODEL＊

- DropTail router : buffer 100 packets
- packet size : 1000 Bytes

connections

- N_{pm} connections using PM
- N_{reno} connections using TCP Reno
  - background traffic

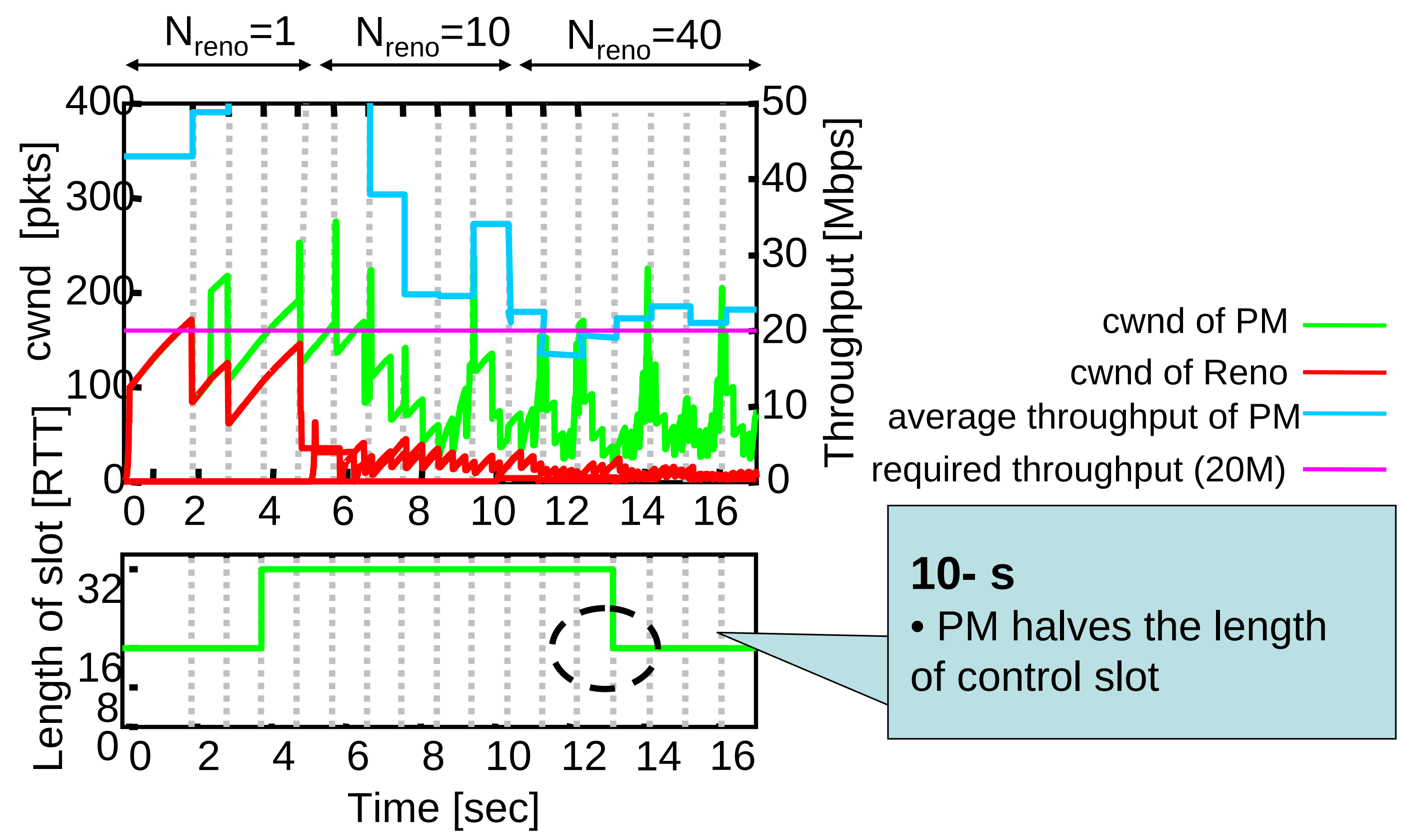
BEHAVIOR EVALUATION

0-5 s

- PM behaves like a TCP Reno connection.

5-10 s

- PM increase the increase degree of cwnd.



10- s

- PM halves the length of control slot

PM can effectively obtain the required throughput
by changing its control according to the network congestion level

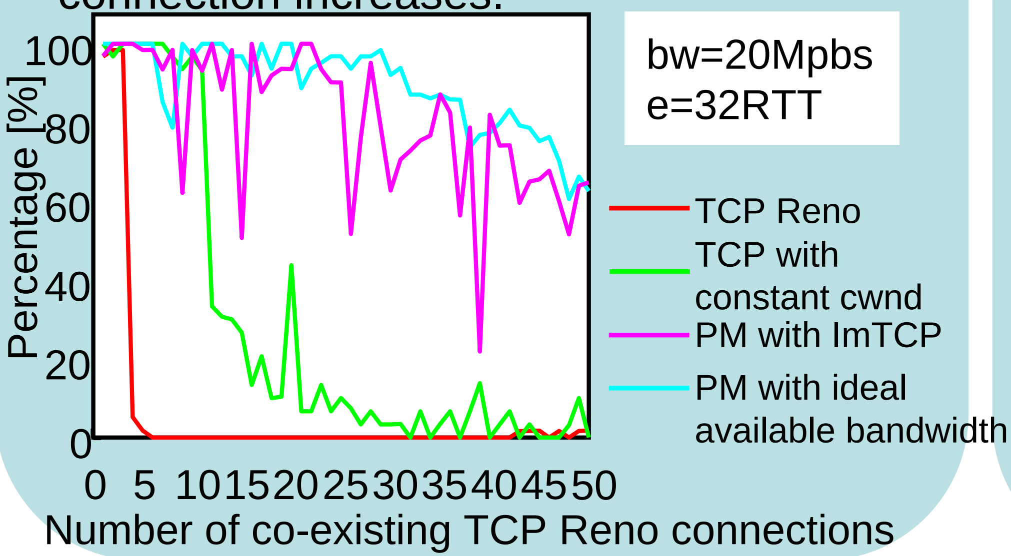
∗ SIMULATION RESULTS ∗

- performance evaluation of proposed mechanism
  - change the number of co-existing TCP Reno connections
  - evaluation index ⇒ $\frac{\text{number of successful evaluation slots}}{\text{number of all evaluation slots}}$

memo:
- TCP with constant cwnd set its cwnd to (srtt * required throughput).
- Ideal available bandwidth is assumed that TCP sender host knows the current information on the available bandwidth.
- PM (N_{pm}=XX) is the proposed mechanism for multiple connections.

CASE OF ONE CONNECTION

- ONLY PM can achieve the required throughput with high probability when the number of the co-existing TCP Reno connection increases.



CASE OF MULTIPLE CONNECTION

(N_{pm}=10)

- PM can achieve the required throughput.
- Larger N can get higher probability.