Performance Trade-Offs of TCP Adaptation Method

Nageswara S. Rao
mputer Science and Mathematics Division
Oak Ridge National Laboratory

Wu-Chun Feng
Computer and Computational Sciences Divi
Los Alamos National Laboratory

International Conference on Networking

August 26-29, 2000, Atlanta, GA

Sponsored by

DARPA Network Modeling and Simulation Program

DOE MICS High-Performance Networking program

ORNL Laboratory Directed R&D Program

National Science Foundation

UT-BATTELLE

line of Presentation

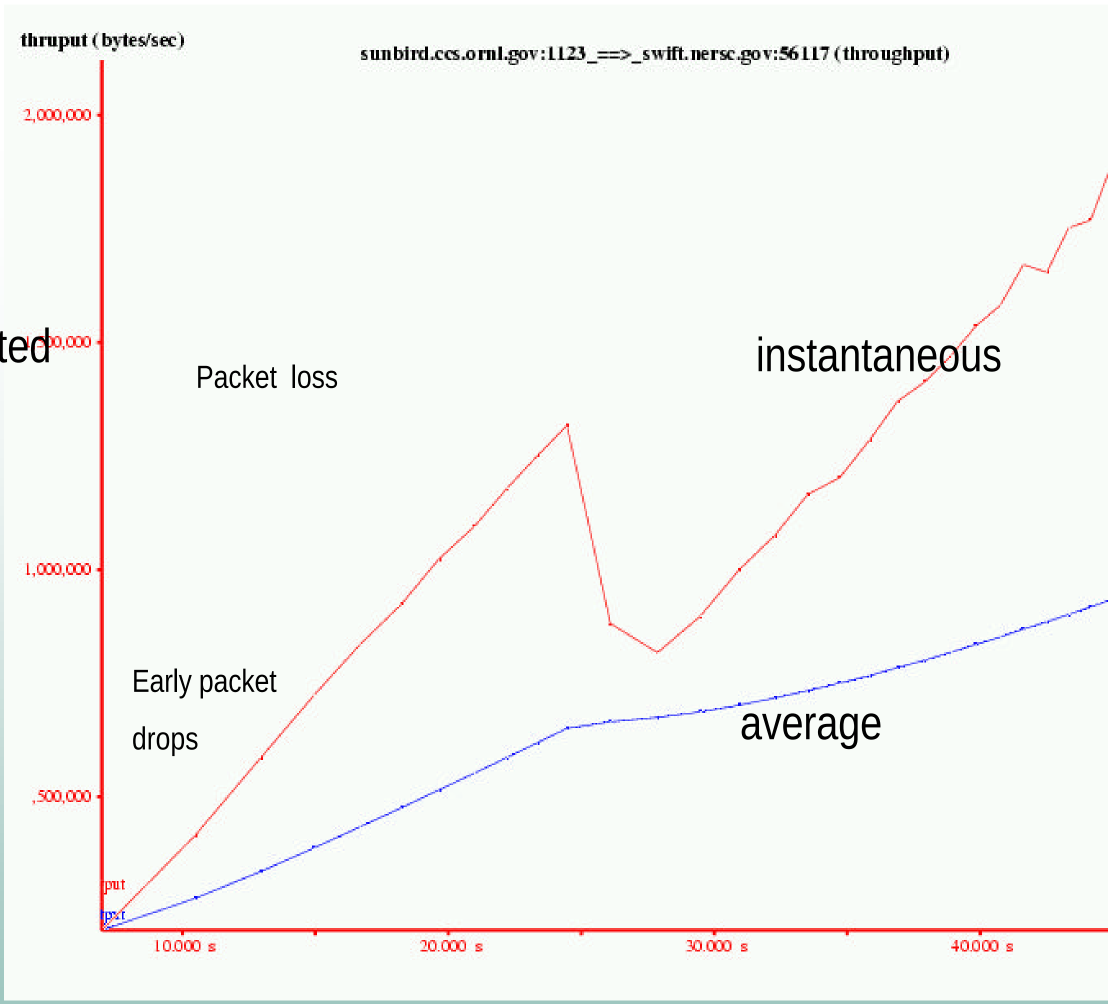
- TCP
  - Performance Issues
  - Simplified Model
- Parallel-TCP
  - Performance Equations
- Comparative Performance
- Dynamic Right-Sizing

ckground

- High bandwidth links ~1Gbps
  - Default TCP stack typically achieves only a fraction of the available bandwidth
- Reasons
  - Inadequately Tuned buffers
    - Dynamic right-sizing (Feng et al)
  - Dynamics of TCP – AIMD (this paper)
    - Early losses prematurely terminate slow-start
- Motivation
  - Just simply using parallel streams improves throughput
    - Understand the mechanism for parallel-TCP
  - When and how to employ these methods
    - SLAC – U Wisconsin: Parallel TCP
    - SLAC – Rice U.: Buffer tuning
    - SLAC – LANL : Combination

Performance – Thanks to Tom Dunigan, ORNL

- osses during startup:
- rt is prematurely terminated
- e BW= 500 Mbps
- d BW= 18M Mbps

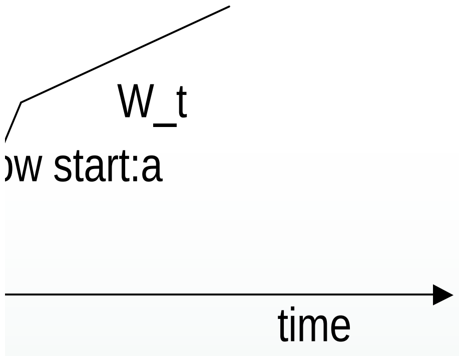


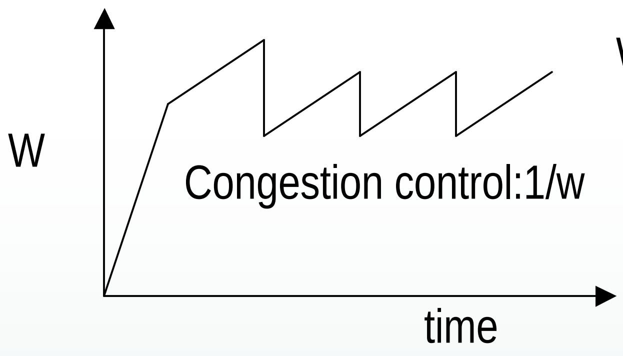
ORNL, TN
LBNL, CA

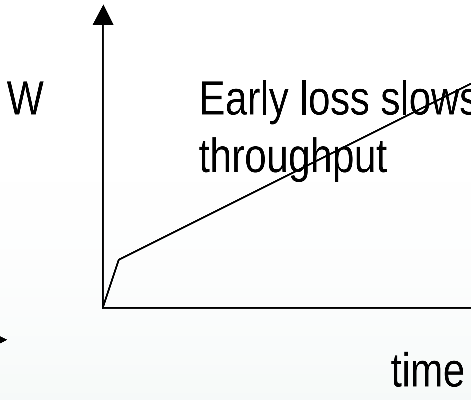
artment of Energy
ge National Laboratory

UT-BATT

ıplified View: Dynamics of TCP







CP Outline

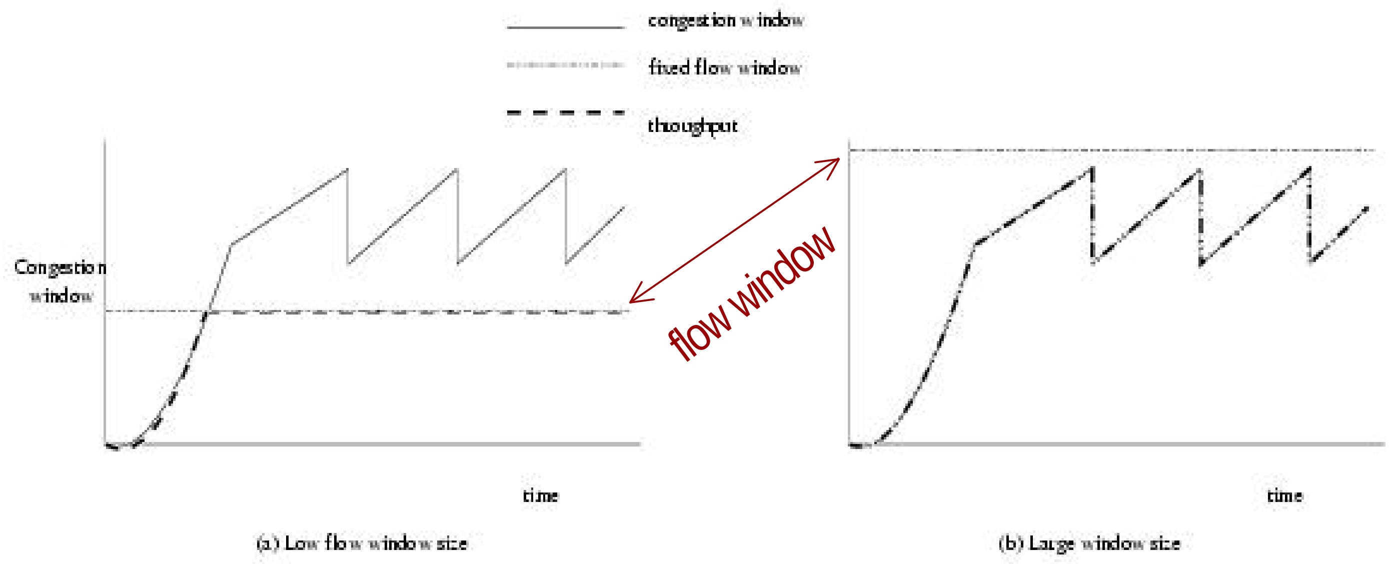
- Uses window mechanism to send W bytes/sec
- Dynamically adjusts W to network and receiver state
  - Keeps increasing is no loses
  - Keeps shrinking if losses are detected
- Slow start phase:
  - W increase exponentially until W_t or loss
- Congestion Control: AIMD
  - linear increase W with delivered packets
  - Multiplicative decrease with loss

v- and high-FS Regions of TCP

FS Region

Slow-start followed by constant flow

Small flow window – no losses



(a) Low flow window size

(b) Large window size

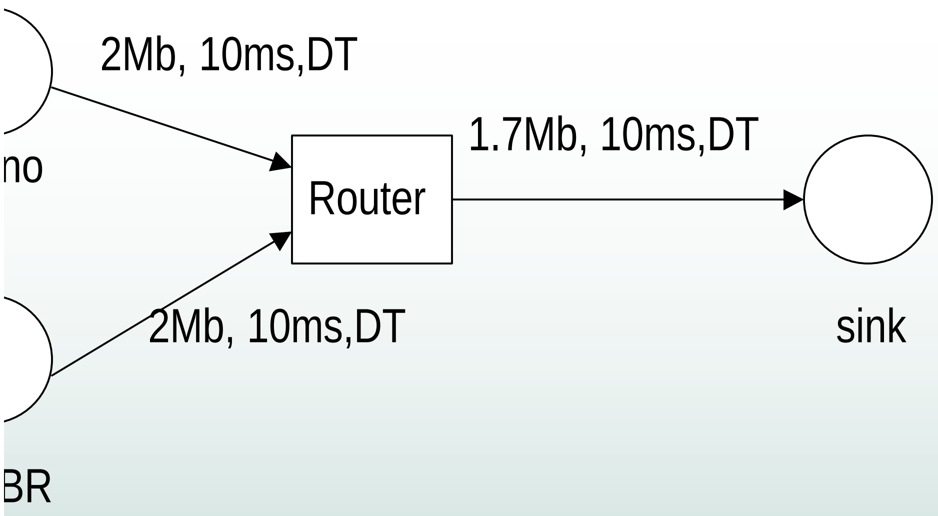
- High-FS Region
  - Slow-start followed by saw-tooth va
  - Low bandwidth or high loss-rate

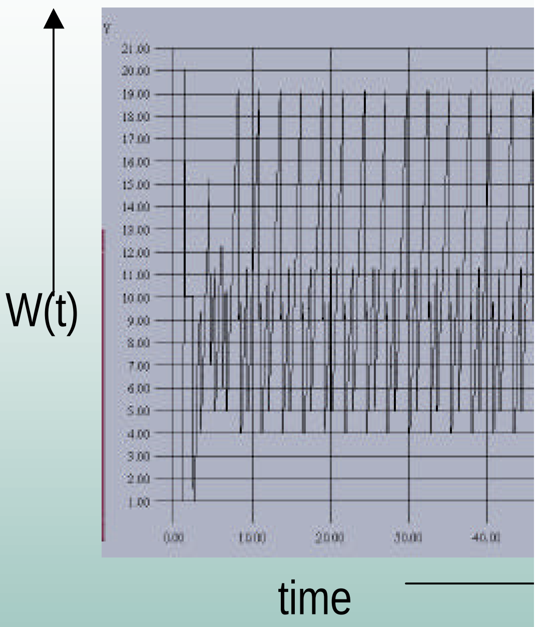
ulation Setup:TCP Competing with UDP

simulation)

CBR rate is varied to control

available bandwidth on the second link

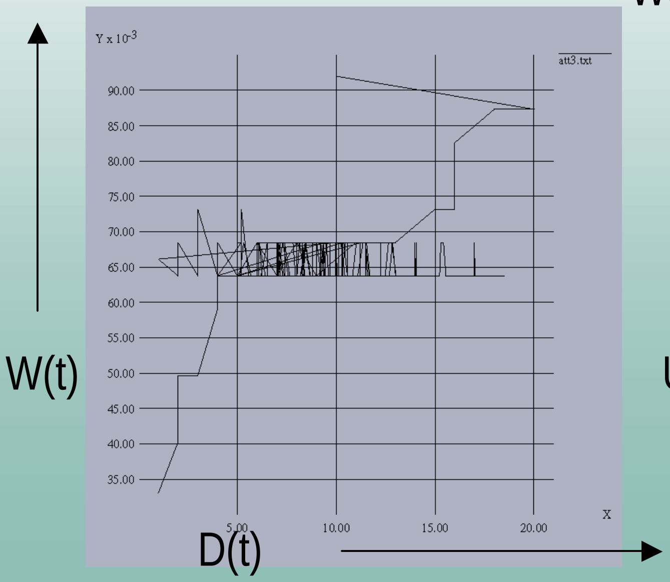




are phase plot:

w-size W(t) vs.

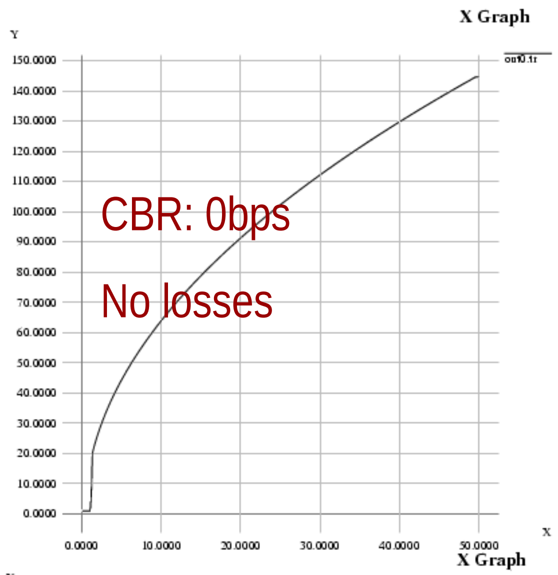
d-to-end delay D(t)

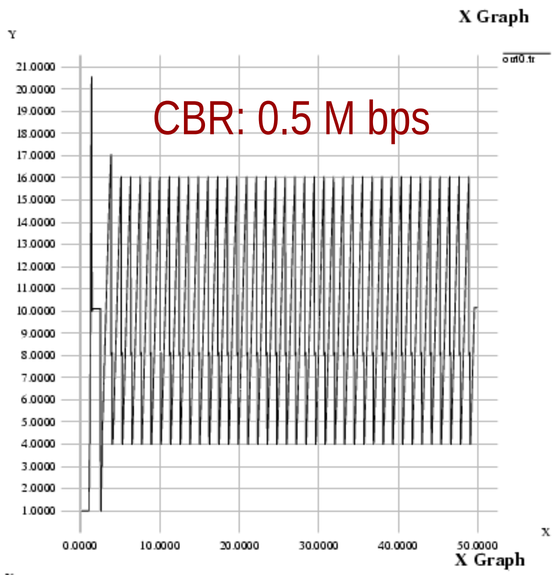


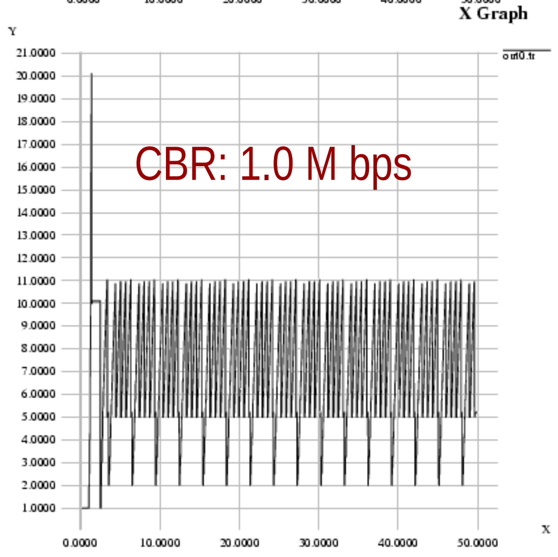
UDP/CBR=1Mbs

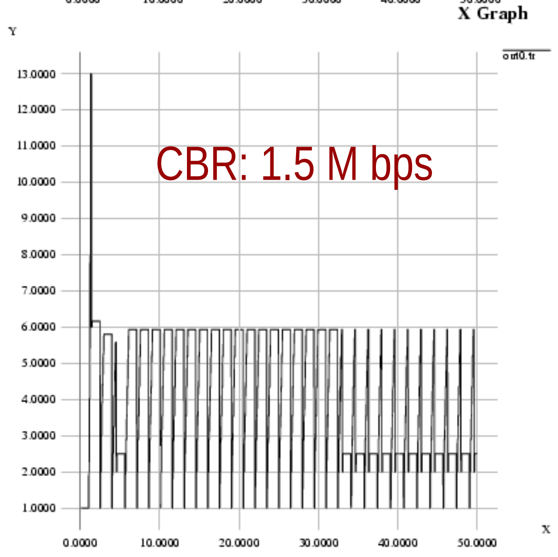
UT-BATT

ulation Results: W(t)-t

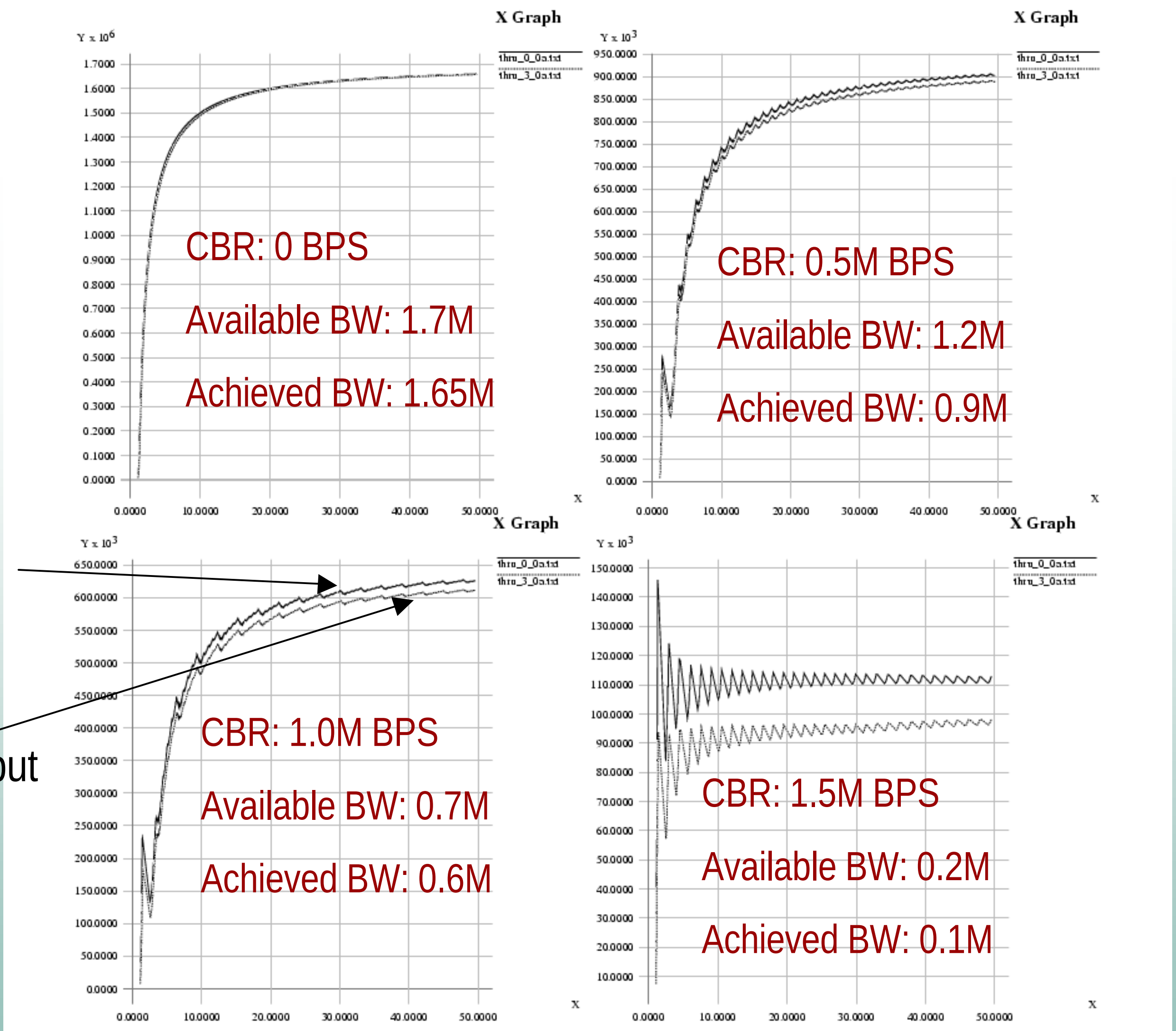








›ughput:



ce output

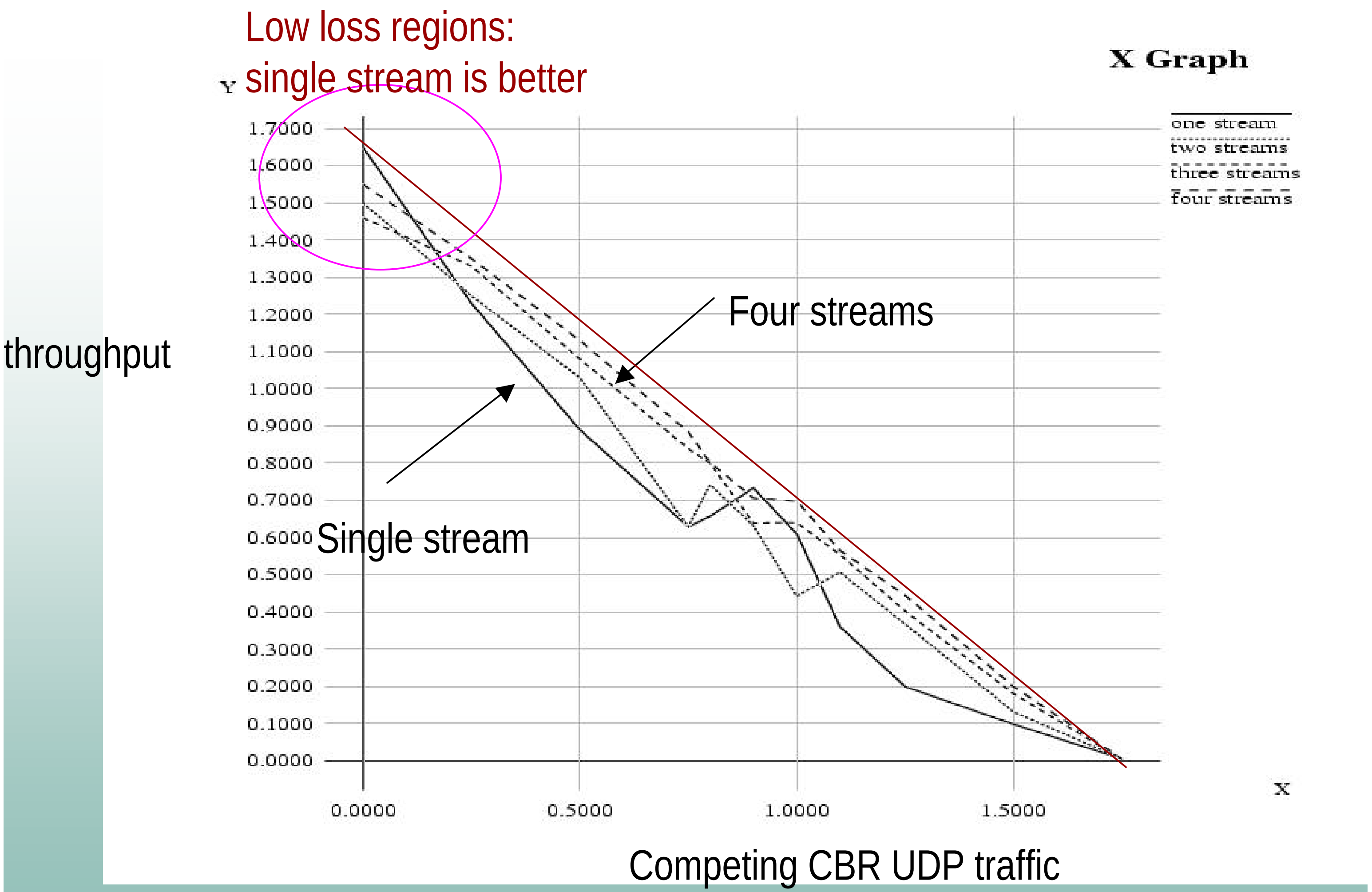
ver goodput

allel-TCP

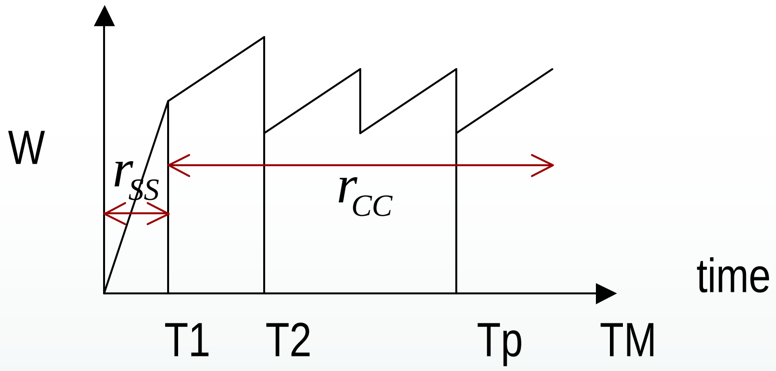
- Method:
  - Divide the message into equal parts
  - Send them as individual streams
- Adhoc Method
  - Developed by application users on >100Mbps networks
  - Easy to use and performs very well in practice – part of GridFTP
  - Typically improves throughput by a
    - mulltiplicative factor in >100Mbps networks
    - Smaller factor over Internet
- Analysis
  - Mostly in congestion-control phase
    - Hacker (2002), Kelly (1999), Crowcroft et al (1998)
  - Slow start phase has not been addressed earlier
    - But has significant effect on throughput
    - Complicated dynamics – due to interacting streams

oughput of Parallel-TCP: Simulation Results:

cally throughput is better if more streams are employed



enario: Sequence of p losses:



implifying Assumption:

- r_{SS} : "average" growth rate of W(t) during slow start phase
- r_{CC}: "average" growth rate of W(t) during congestion control phase

- $r_S(t)$: growth rate of single stream

$$r_S(t) ? r_{SS}U_{[T0,T1]}(t) ? r_{CC}U_{(T1,TM]}(t)$$

vhere $U_I(t) ? 1$ if $t ? I$

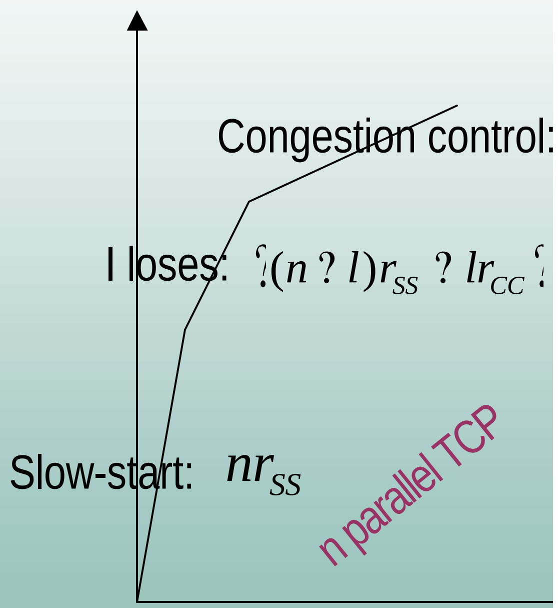
$U_I(t) ? 0$ otherwise

allel-TCP: Window Growth-Rate

Growth rate n-parallel TCP is

$$r_P(t) ? nr_{SS}U_{[T0,T1)} ? (n ? 1)r_{SS} ? r_{CC} U_{[T1,T2)} ? \ldots ? nr_{CC}U_{[Tn,TM]}$$

ime interval	Single TCP	n-parallel TCP
[T0,T1)	r_{SS}	nr_{SS}
[T1,T2)	r_{CC}	$(n ? 1)r_{SS} ? r_{CC}$
[Tl,Tl+1)	r_{CC}	$(n ? l)r_{SS} ? lr_{CC}$
[Tn-1,Tn)	r_{CC}	$r_{SS} ? (n ? 1)r_{CC}$
[Tn,TM]	r_{CC}	nr_{CC}



allel-TCP: Slow Start Phase

Growth rate n-parallel TCP is

$$r_P(t) \; ? \; nr_{SS}U_{[T0,T1)} \; ? \; (n ? 1)r_{SS} \; ? \; r_{CC} \; U_{[T1,T2)} \; ? \; ... \; ? \; nr_{CC}U_{[Tn,TM]}$$

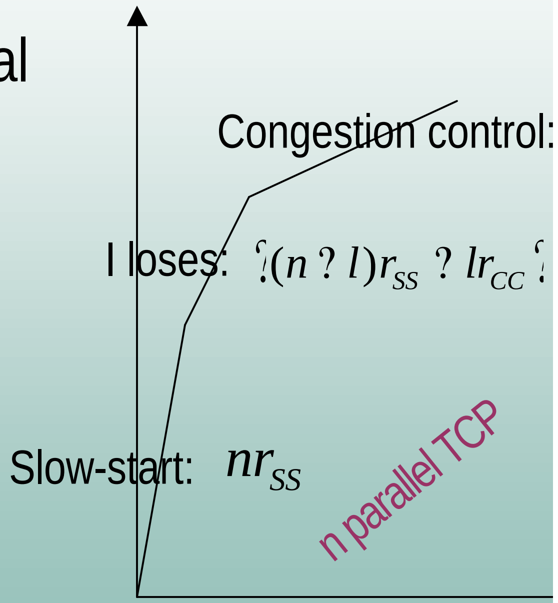
Single vs. n-parallel TCP

– Faster slow start: duration ~c log(W_t)
  - Single: r_{SS}
  - Parallel: $(n ? l)r_{SS} \; ? \; lr_{CC}$

– Sustained slow-start under transient initial loses – throughput grows faster longer
  - Single – small loss spike kills slow start [T0,T1]
  - Multiple – with l spikes, residual rate [T0,Tn]

Summary

Parallel-TCP starts with faster rate and gradually slows down in presence of losses



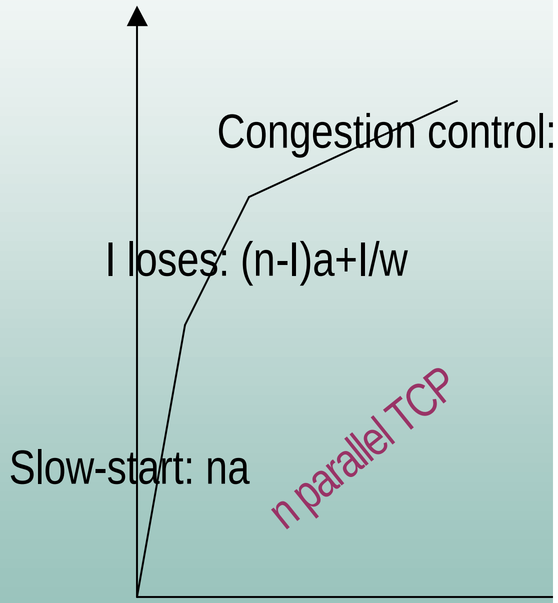
llel-TCP in Congestion Control Phase

aster recovery in Congestion Control

- Single: 1/w
- Parallel: n/w

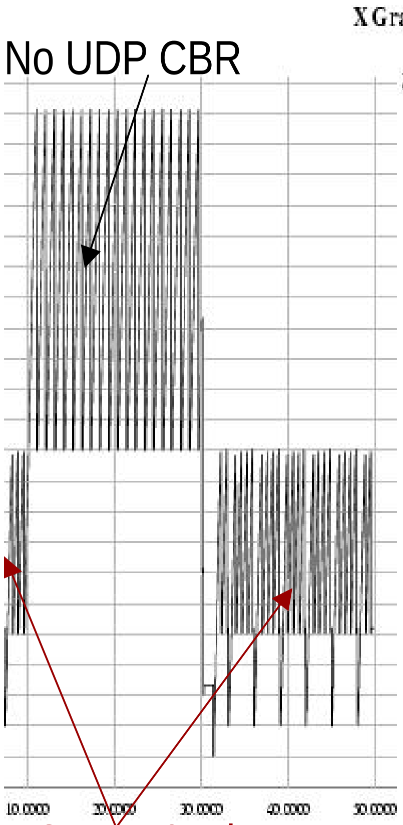
ynamics are very complicated since paths are restricted to a nall set – the streams compete with themselves

- This is the most analyzed phase in past works



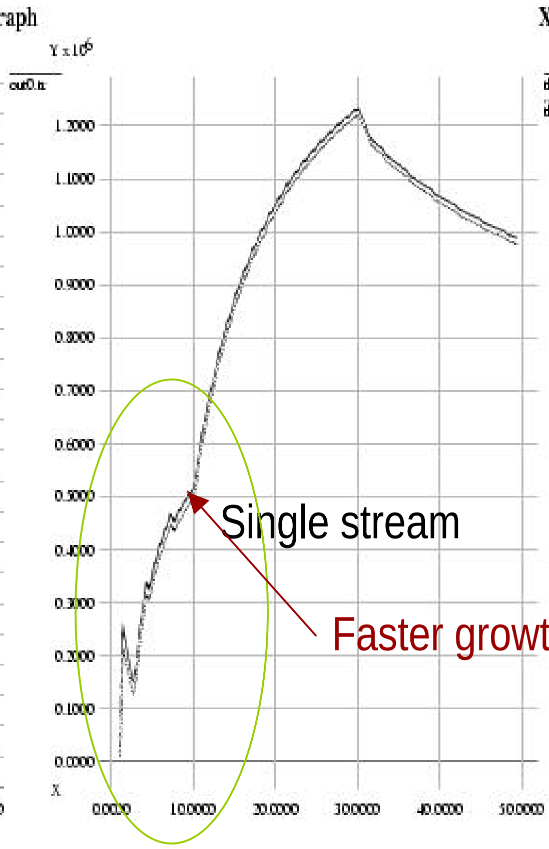
er Performance of Parallel-TCP

ker response and higher throughput



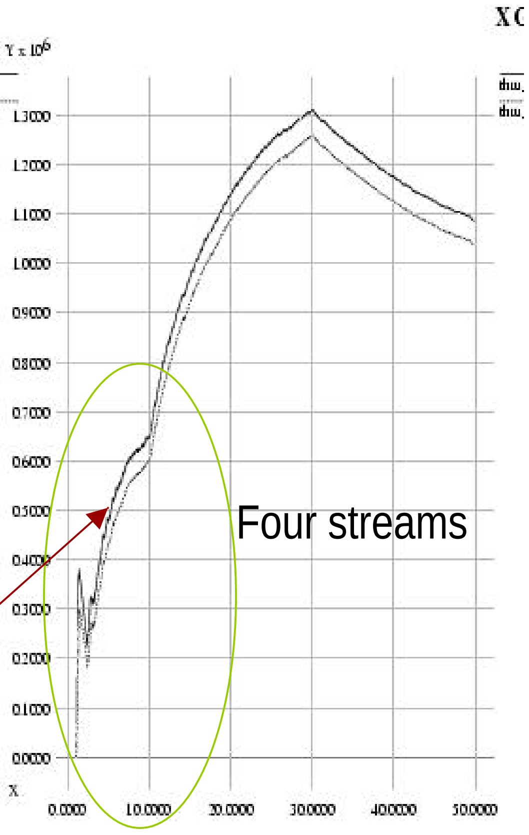
No UDP CBR

P CBR 1.0 Mbps



Single stream

Faster growth



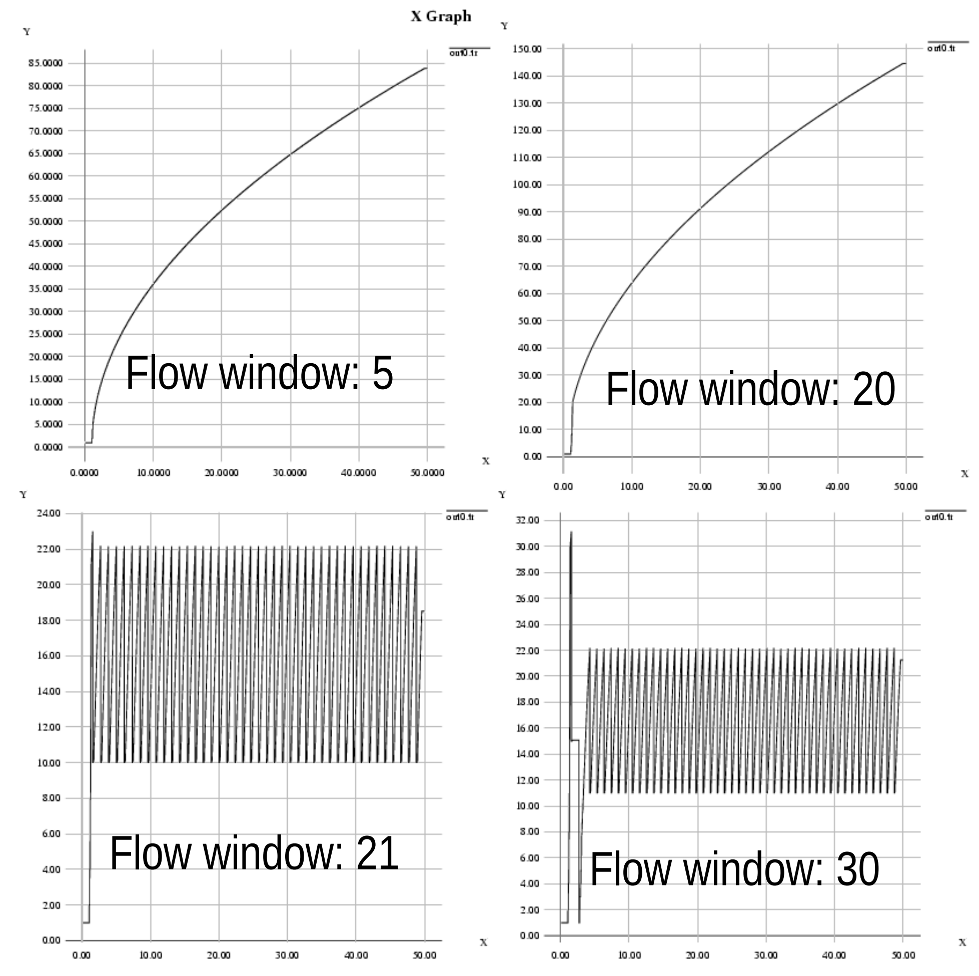
Four streams

ects of Flow Window size
ulation Results

g higher flow
does not mean
nroughput:

has no losses

incurs loses



ects of Flow Window Size

er flow window:

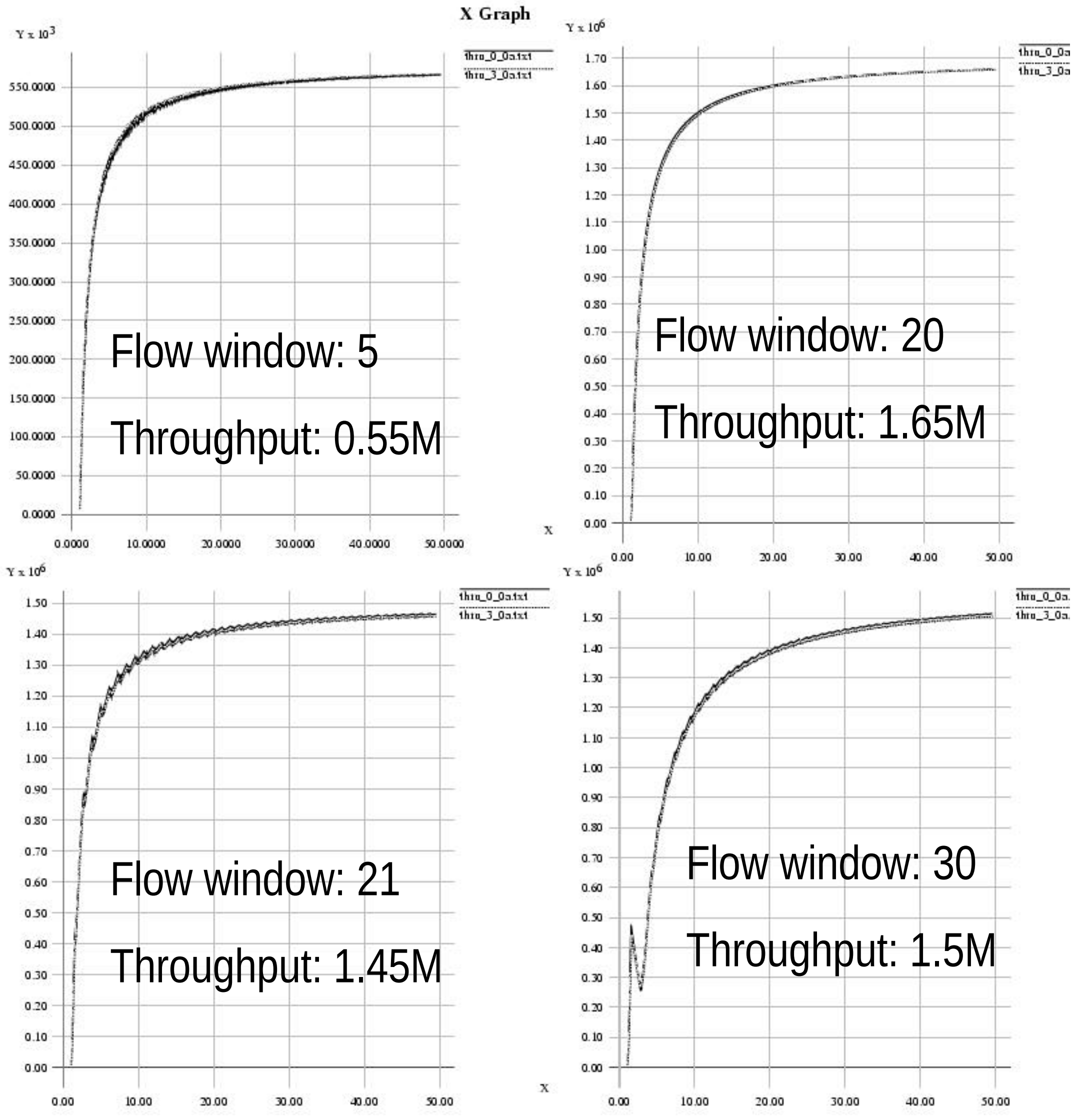
ed AIMD process
h reduces throughput

egy:

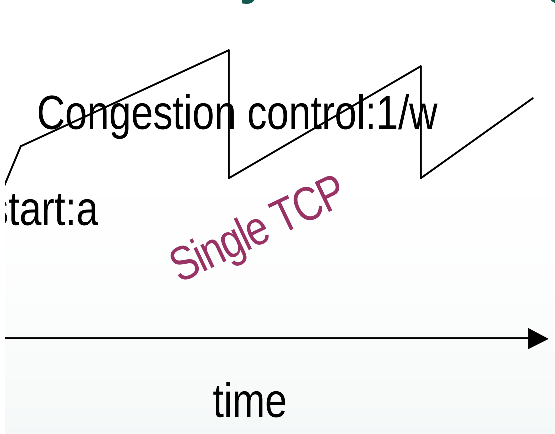
flow window below
eneck bw

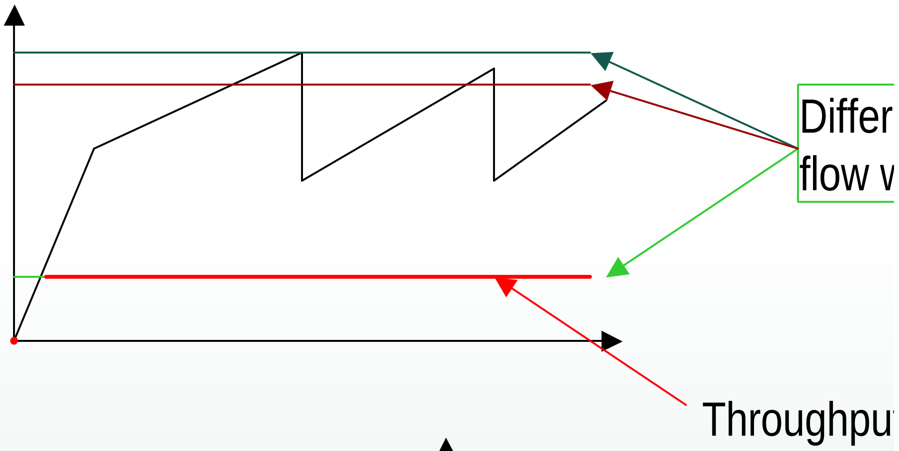
eat:

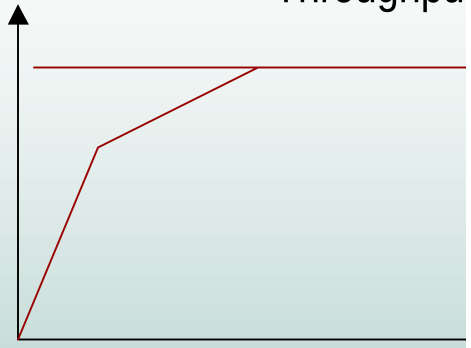
eneck bw may be
e to zero if there is a
peting TCP stream

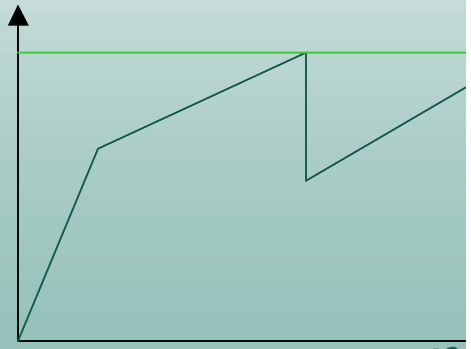


ırse Analysis of Congestion and flow window controls







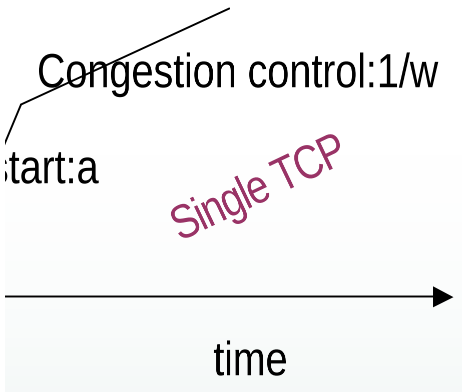


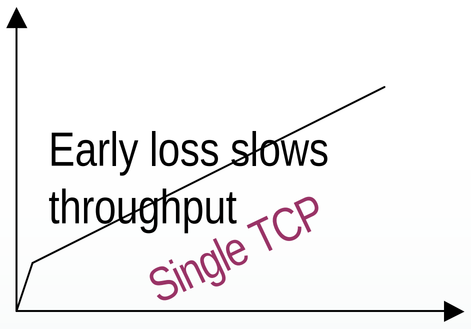
- ow window has significant effect on throughput
  - Non-monotonic relationship between flow-window size and throughput
- ynamic flow-windows vs. n-parallel TCP: Performance depends on sses
  - Low loss – dynamic right sizing is better
    - Choose flow window slightly lower than bottleneck bandwidth
    - Parallel TCP creates additional loses which reduces throughput
  - High loss – parallel-TCP is better
    - Effect of flow window is nullified – essentially single TCP
    - Advantages are same as the single vs parallel TCP

nclusions

- TCP is sub-optimal in high bandwidth links
  - Buffer tuning and parallel streams provide some solution
  - We provide fairly coarse analysis of both methods
    - Parallel-TCP provides better throughput under high loss
      - But fairness issues are unclear
    - Flow-window tuning improves throughput under low loss
      - Degenerates to single stream under high loss
- Several Open Issues
  - Detailed analysis – employ actual rates r(t)
  - Dynamics of window sizes and packet delays
  - General Fairness Issues

ırse Analysis of Initial Dynamics of Parallel-TCP

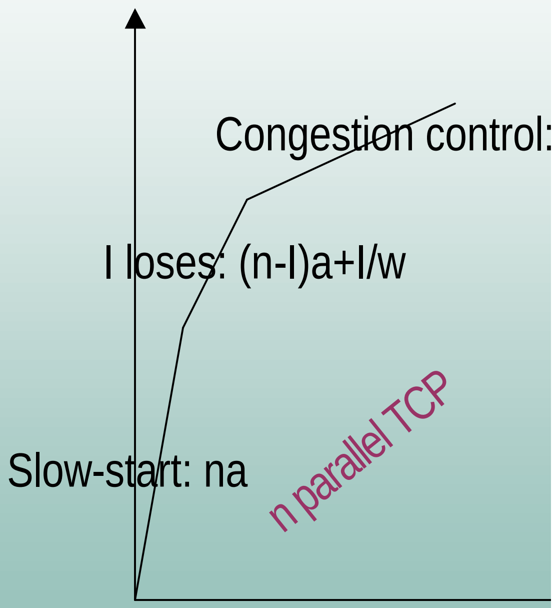




Kelly (2001) result deal
congestion control pha

ingle vs. n-parallel TCP

- Faster slow start: duration ~c log(W_t)
  - Single: a
  - Parallel: na
- Sustained slow-start under transient initial loses – throughput grows faster longer
  - Single – small loss spike kills slow start
  - Multiple – with l spikes, residual rate (n-l)a+l/w
- Faster recovery in Congestion Control
  - Single: 1/w
  - Parallel: n/w



ynamics are very complicated since paths are restricted to a small set
he streams compete with themselves