TCP Westwood with Agile Probing:

Handling Dynamic Large Leaky Pipes

Problem Definition

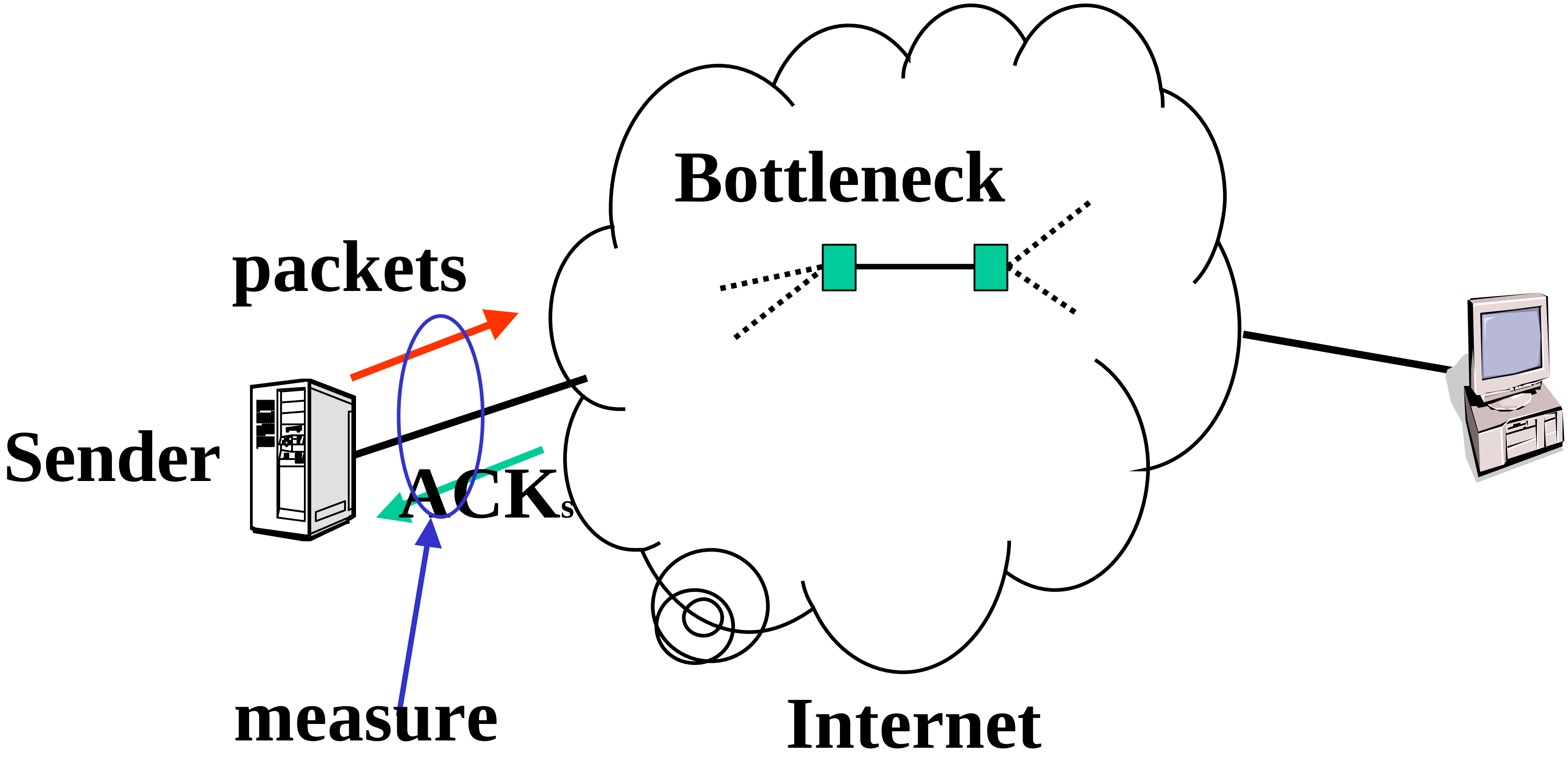
- Leaky Pipes: packet loss due to error
  - Unjustified *cwnd* cut and premature Slow Start exit
- Large Pipes: Large capacity and long delay
  - Control scheme may not scale
- Dynamic Pipes: Dynamic load/changing link bandwidth (Due to change of technologies, e.g., 802.11, Bluetooth, 1XRTT)
  - Linear increase limits efficiency

Key Solution Components

Sender-side only enhancement:

- TCP Westwood
- Persistent Non-Congestion Detection:
  - Detect extra unused bandwidth
  - Invoke Agile Probing
- Agile Probing: Probe efficiently but not too fast

TCP Westwood (TCPW)

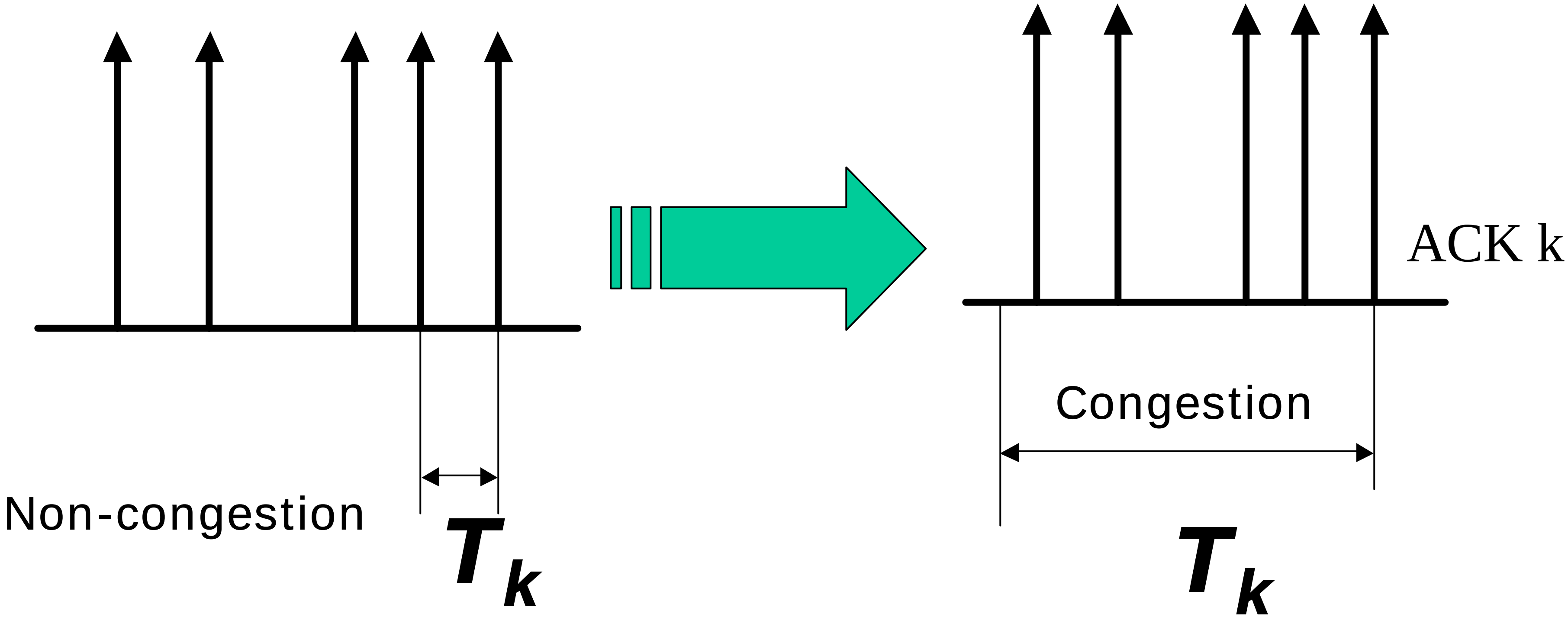


TCP Westwood (TCPW)

- Network viewed as blackbox;Estimation done on sender
- After dup-acks:
  - *cwnd* and *ssthresh* ← ERE * RTTmin
- After a timeout:
  - *ssthresh* ← ERE * RTTmin, *cwnd* ← 1

Eligible Rate Estimate (ERE)

ERE Adaptation:



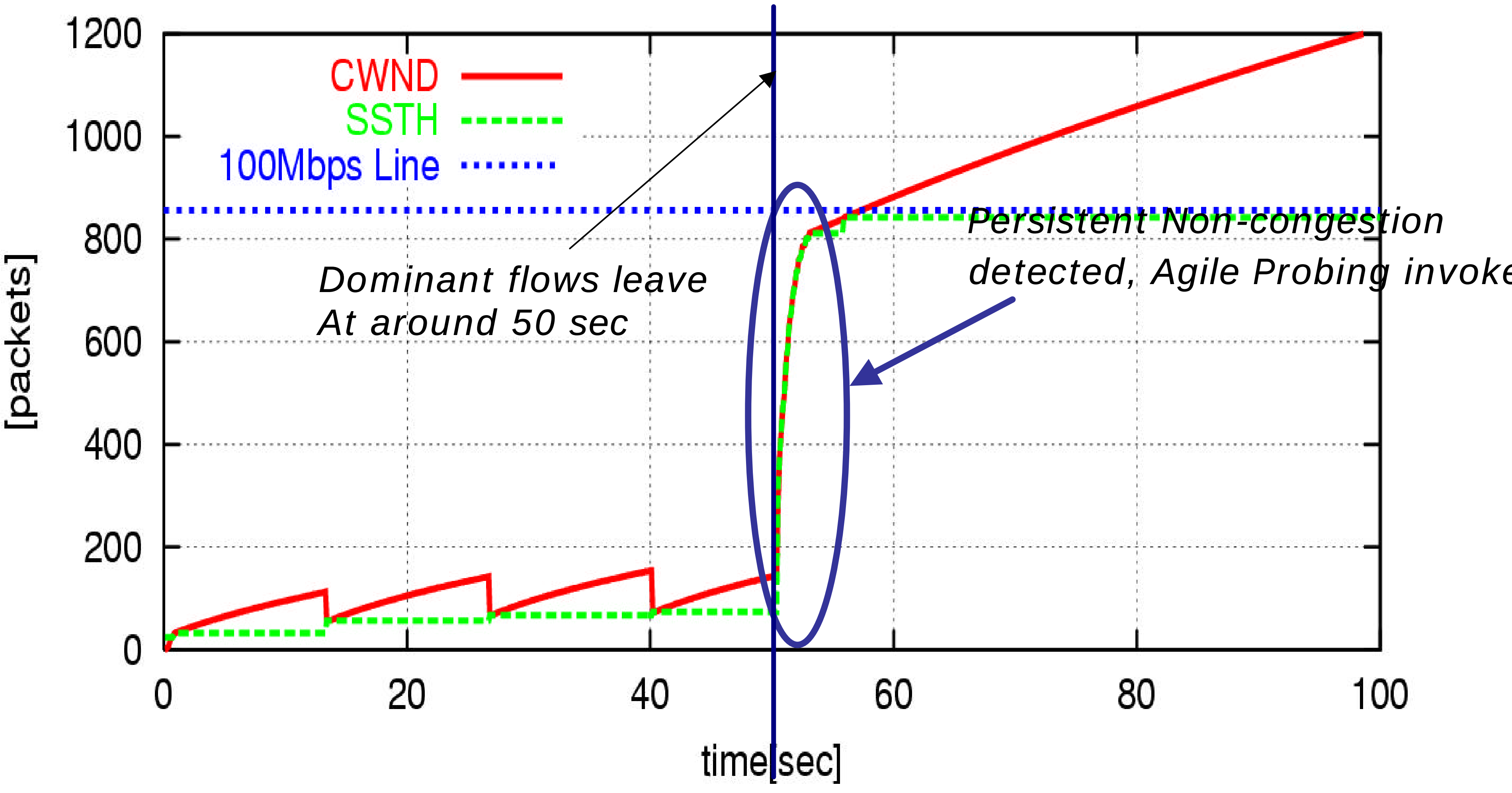
Eligible Rate Estimate (ERE)

- ERE sample:Calculated by bytes delivered in interval T_k
  - Congestion level decided by expected rate and achieved rate
  - Light Congestion: short Tk,(packet-pair like)
  - Heavy Congestion: long Tk, (packet-train like)
- Using discrete low pass filter to get smoothed ERE

Persistent Non-Congestion Detection(PNCD)

- Objective: Detect unused bandwidth/invoke Agile Probing
- Observe Achieved Rate (AR) and Expected Rate (ER)
- If AR follows ER for a considerably long time -> ***PNC*,** indicating extra unused bandwidth->Agile Probing invoked

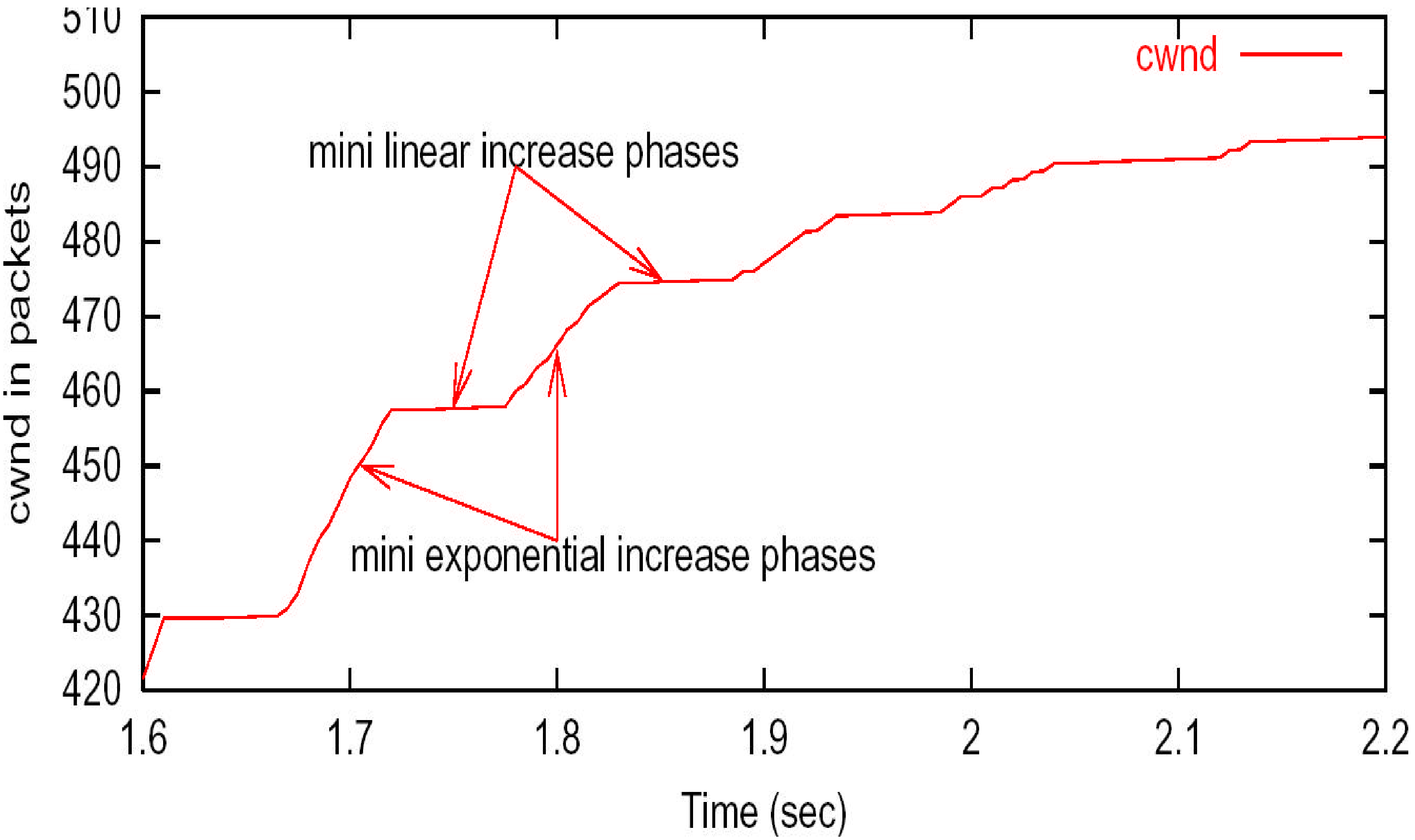
Persistent Non-Congestion Detection(PNCD)



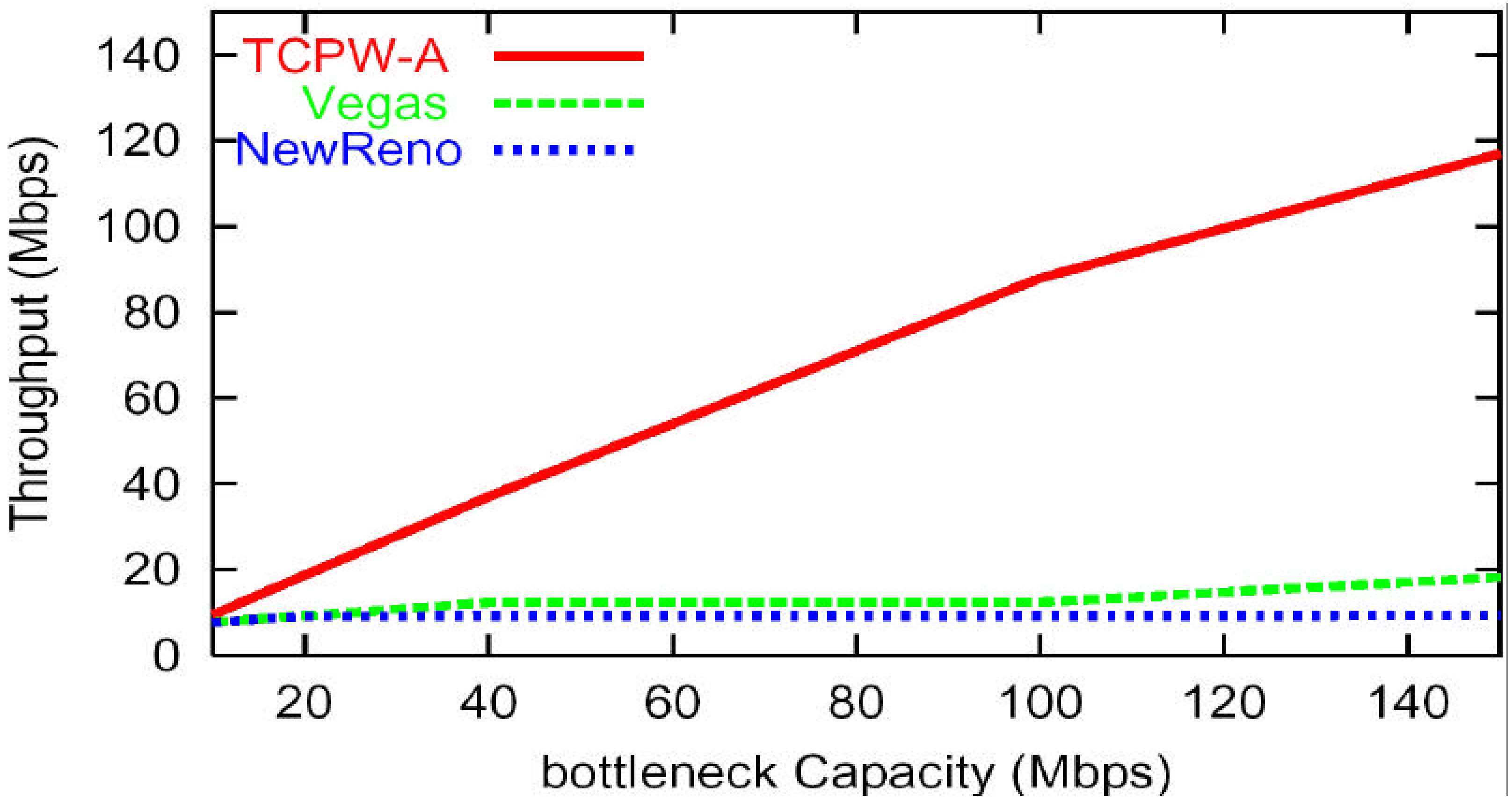
Agile Probing

- Objective: Guided by ERE, converge *faster* to *more appropriate* ssthresh
  - adaptively and repeatedly resets *ssthresh* to ERE*RTTmin
  - Exponentially increase cwnd if ssthresh >cwnd
  - Linearly increase cwnd if ERE < ssthresh
  - Exit Agile Probing when packet loss is detected

Agile Probing

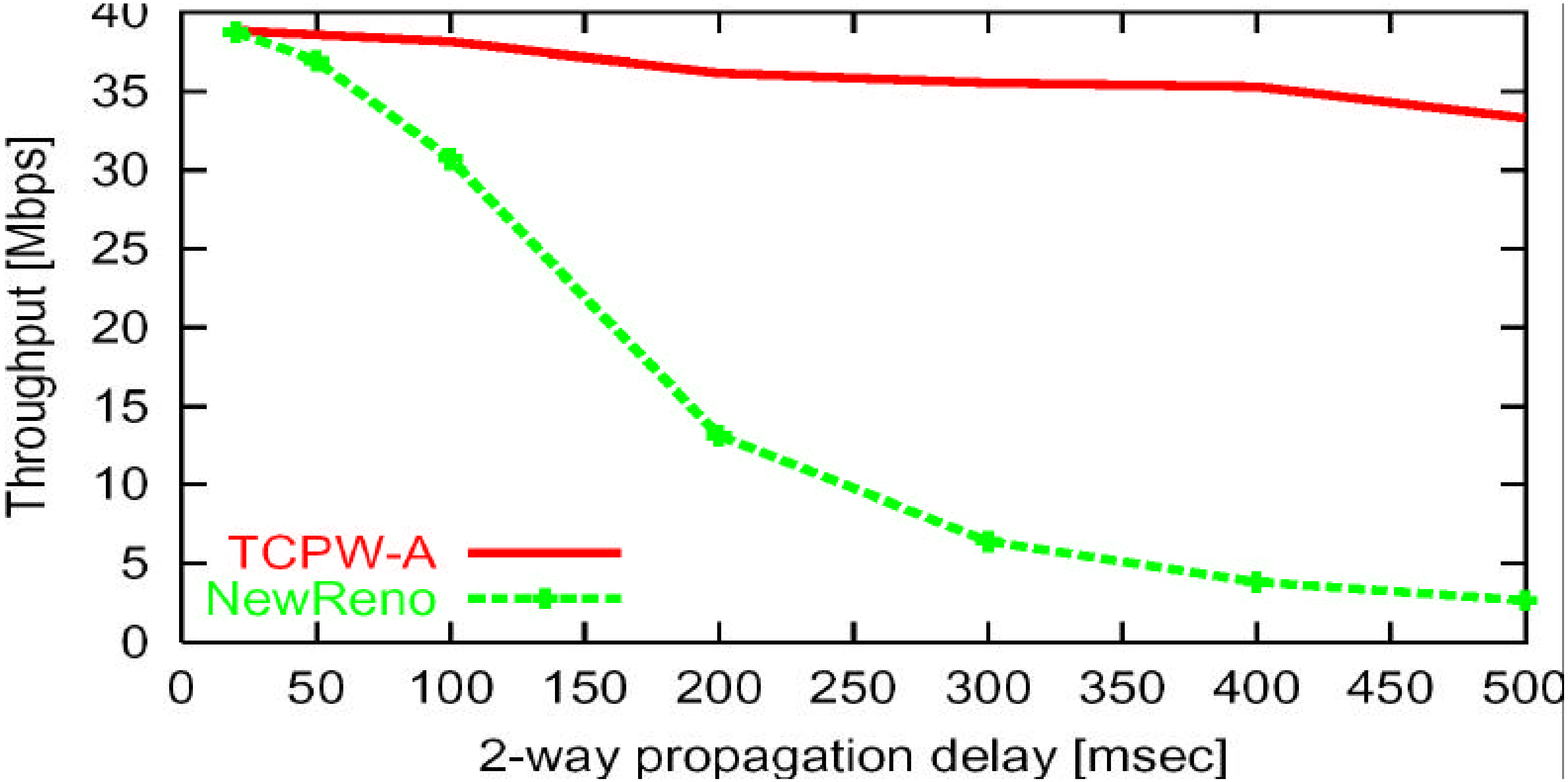


Performance Evaluation (1)



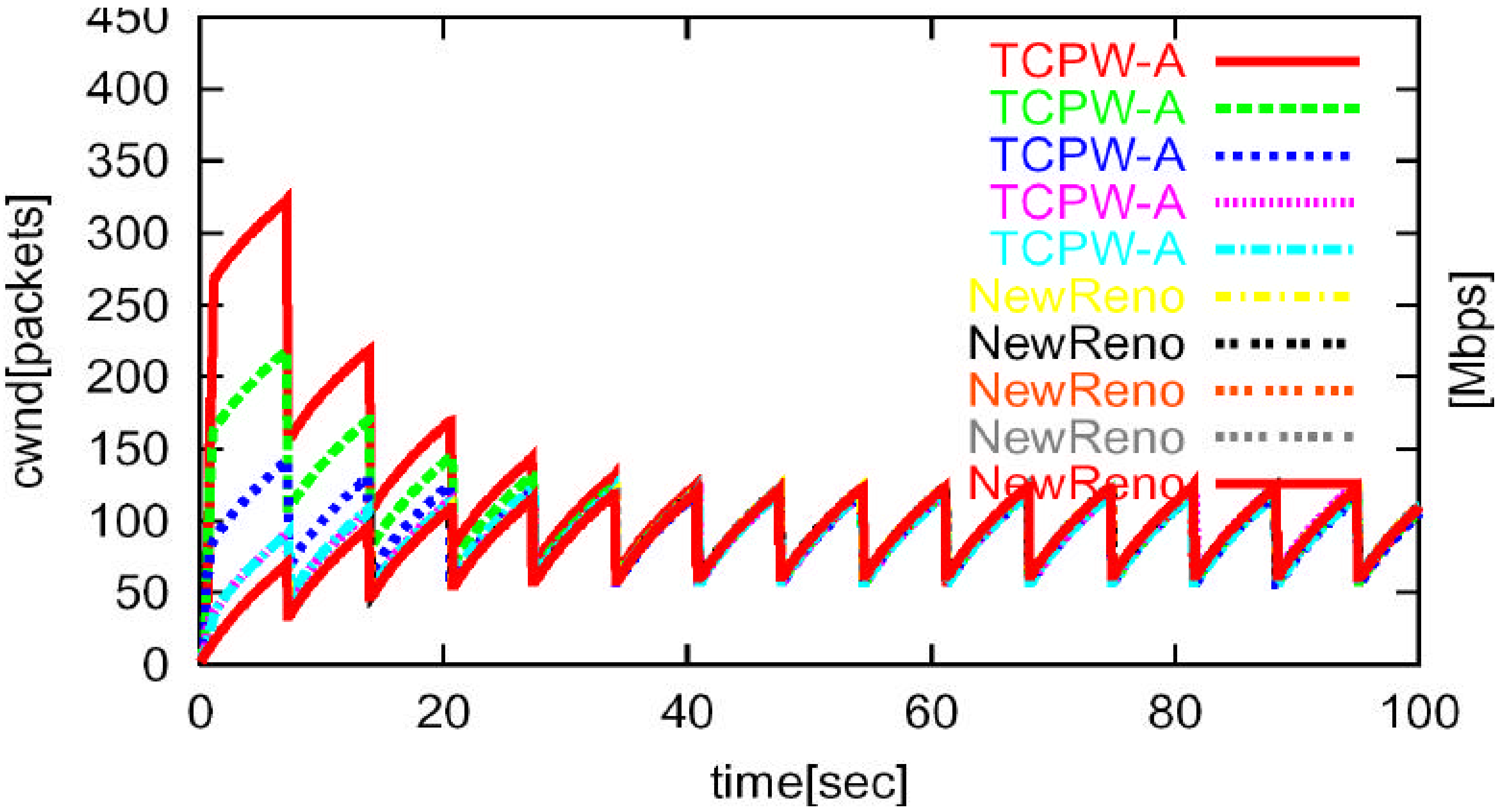
Throughput vs. bottleneck capacity during first 20 seconds (RTT=100ms)

Performance Evaluation (2)



Throughput vs. delay: 100 flows (each last 30sec) randomly spread out during 20 minutes (bottleneck capacity = 45Mbps)

Performance Evaluation (3)



Friendliness and convergence