TCP Veno

TCP Enhancement for Transmission Over Wireless Access Networks

C. P. Fu and S. C. Liew

IEEE JSAC, 2003

Presented by J. L. Chiang

July 10, 2003

Outline

- Introduction
- TCP Veno
- Simulation Study
- Conclusions

Introduction

- Conventional TCP treats packet loss as congestion notification, however, this cannot apply in error-prone wireless networks.
- TCP (Reno) also suffers from heteroge-neity on the internet.
  - Reliability: optical v.s. wireless
  - Asymmetry: ADSL
  - Propagation delay: satellite v.s. terrestrial

TCP Vegas

- Proactive congestion control

$$Expected = cwnd / BaseRTT$$

$$Actual = cwnd / RTT$$

$$Diff = Expected - Actual$$

Let *N* be the backlog at bottleneck link:

$$RTT = BaseRTT + N / Actual$$

$$N = Actual * (RTT - BaseRTT) = Diff * BaseRTT$$

- Use *N* to adjust TCP window size to avoiding packet loss due to buffer overflow.
- 37%~71% improvement over Reno

Problems of Vegas

- Vegas cannot coexists with Reno, since it is less aggressive than Reno's policy, which continues to increase window size until packet loss occurs.

- The measured backlog is not necessary the data backlog in asymmetric networks.
  - Underutilization on the wider forward path

TCP Veno

- Veno= **Ve**(gas)+(Re)**no**
- Veno use the measurement of *N* not as a way to adjust window size proactively, but rather as an indication of whether the connection is in a congestive state.
  - $N<\beta$: random loss
  - $N>\beta$: congestive loss
- Only the sender stack is modified.
- With 1% random packet loss rate, improvement up to 80% is achieved.

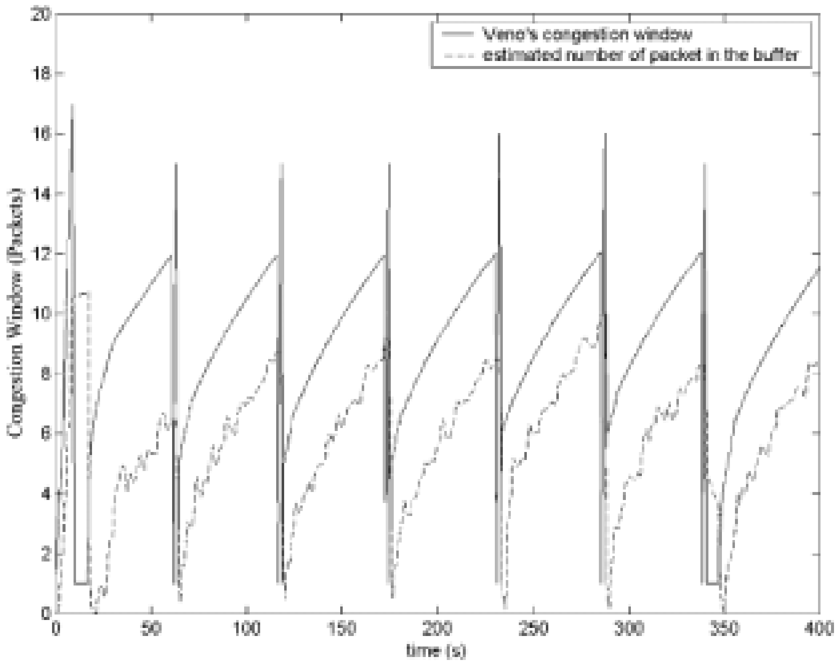
Modified Additive Increase

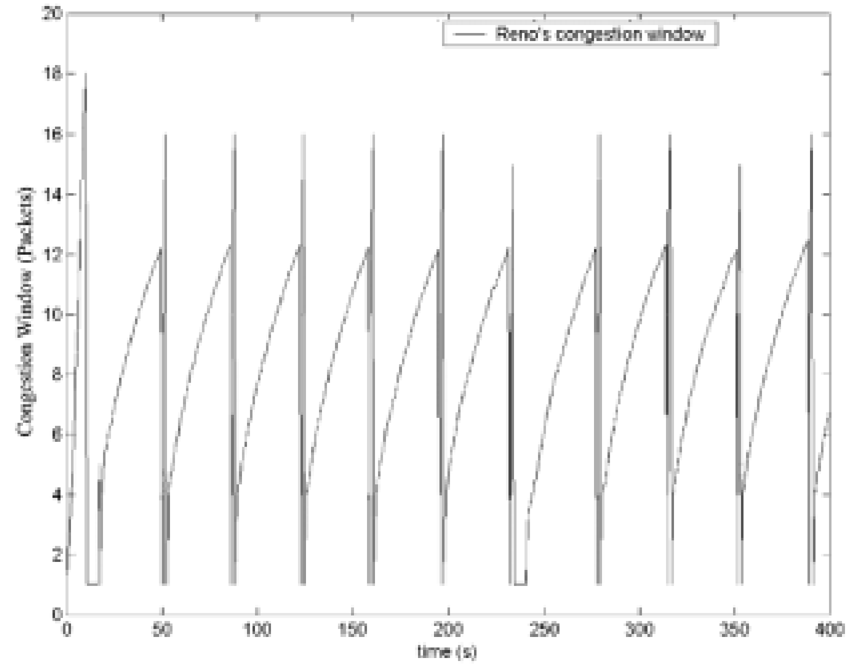
- Slow Start (unchanged)
- Additive Increase:

If $(N < \beta)$ // available bandwidth not fully utilized
  set $cwnd = cwnd + 1/cwnd$ when each new ACK is received
else if $(N \geq \beta)$ // available bandwidth fully utilized
  set$cwnd = cwnd + 1/cwnd$ when *every other* new ACK is received

Modified Additive Increase

- Fewer oscillations
- Longer large-window region
- Less congestion loss





Modified Multiplicative Decrease

- Timeout (unchanged)

  1)ssthresh=cwnd/2;

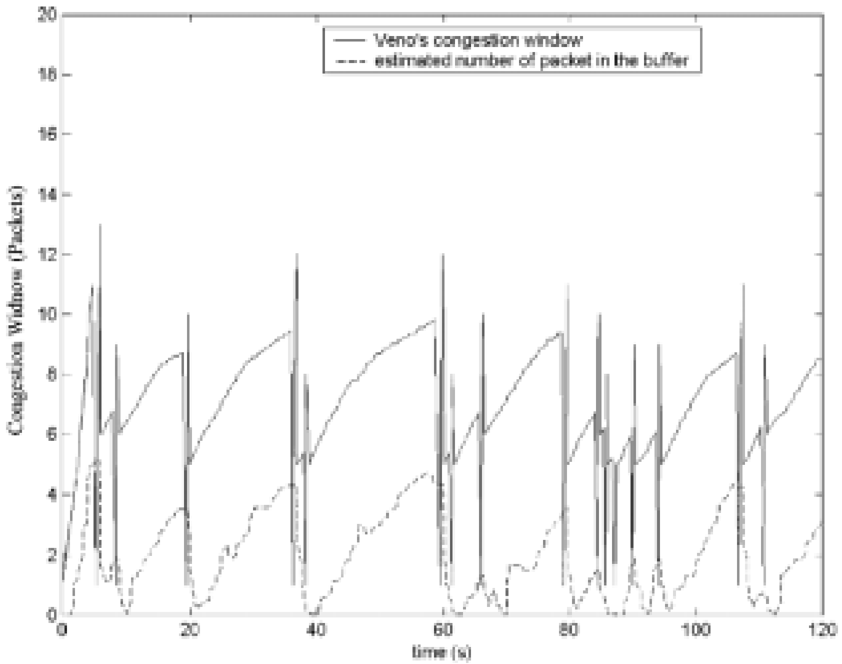
  2)cwnd=1;

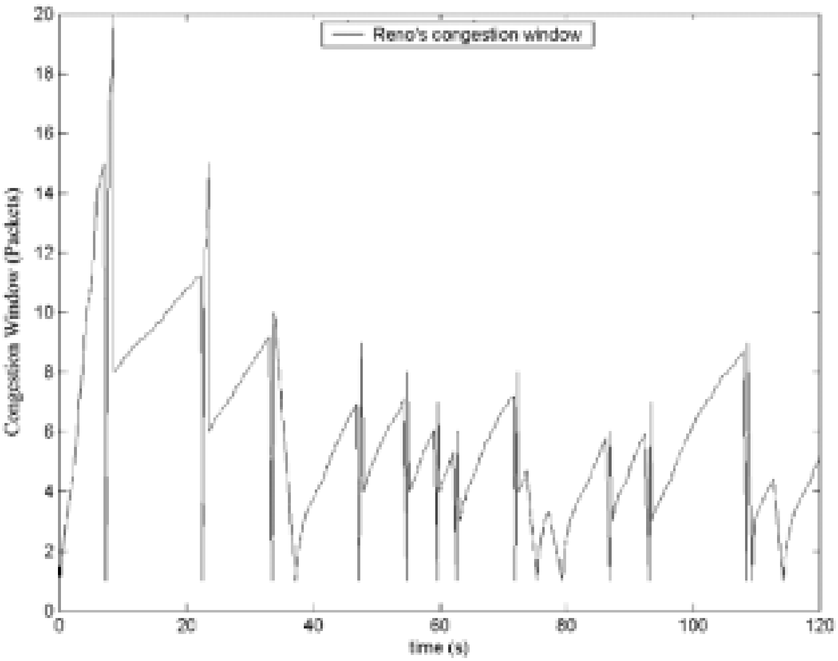
- Fast retransmit

  if $(N < \beta)$ // random loss due to bit errors is most likely to have occurred
  $ssthresh = cwnd * (4/5)$;
  else $ssthresh = cwnd/2$; // congestive loss is most likely to have occurred

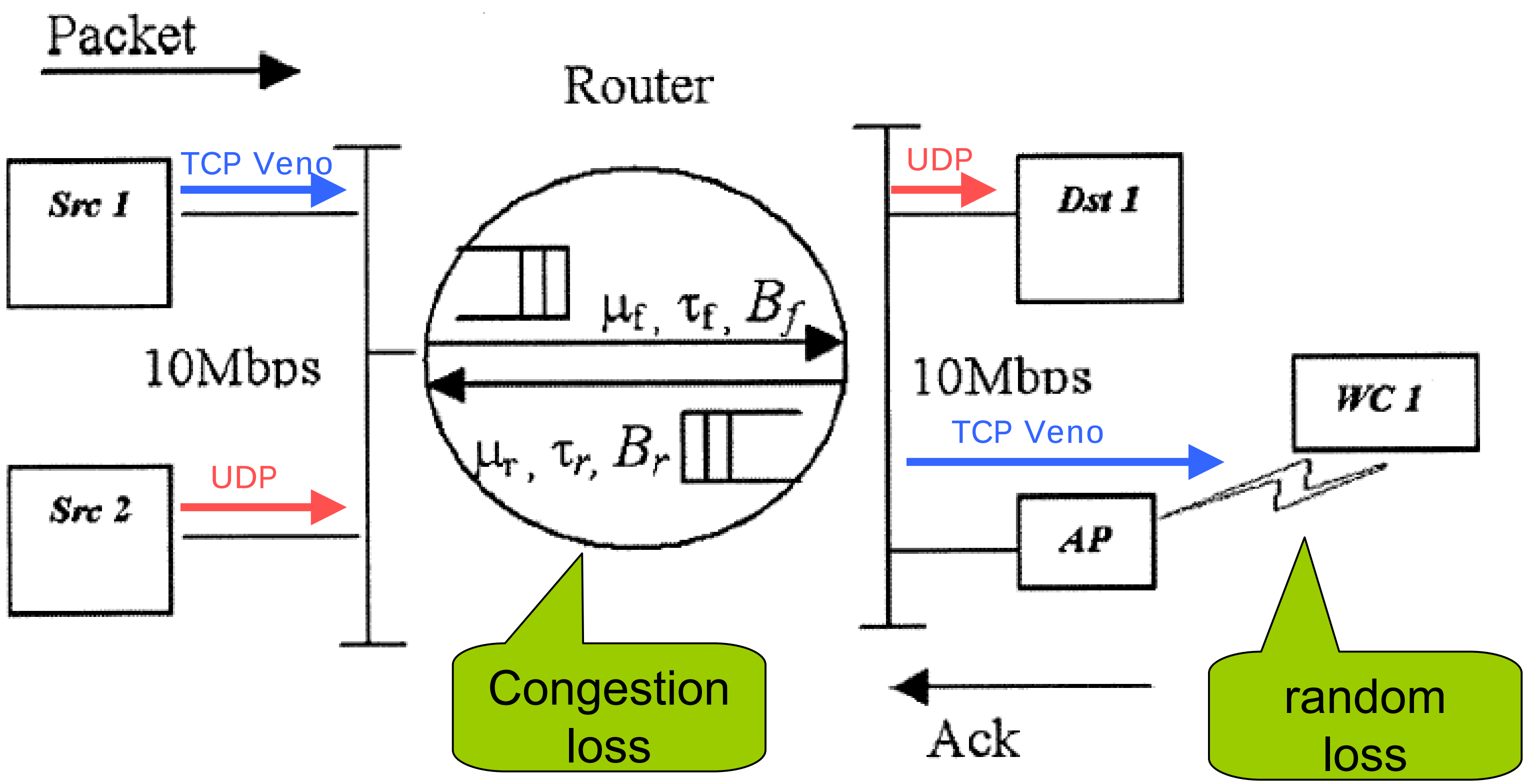
  1) Retransmit the missing packet
     set $ssthresh = cwnd/2$
     set $cwnd = ssthresh + 3$.
  2) Each time another dup ACK arrives, increment *cwnd* by one packet.
  3) When the next ACK acknowledging new data arrives, set *cwnd* to *ssthresh* (value in step 1).

Modified Multiplicative Decrease





Simulation A: Distinguishing Packet Loss

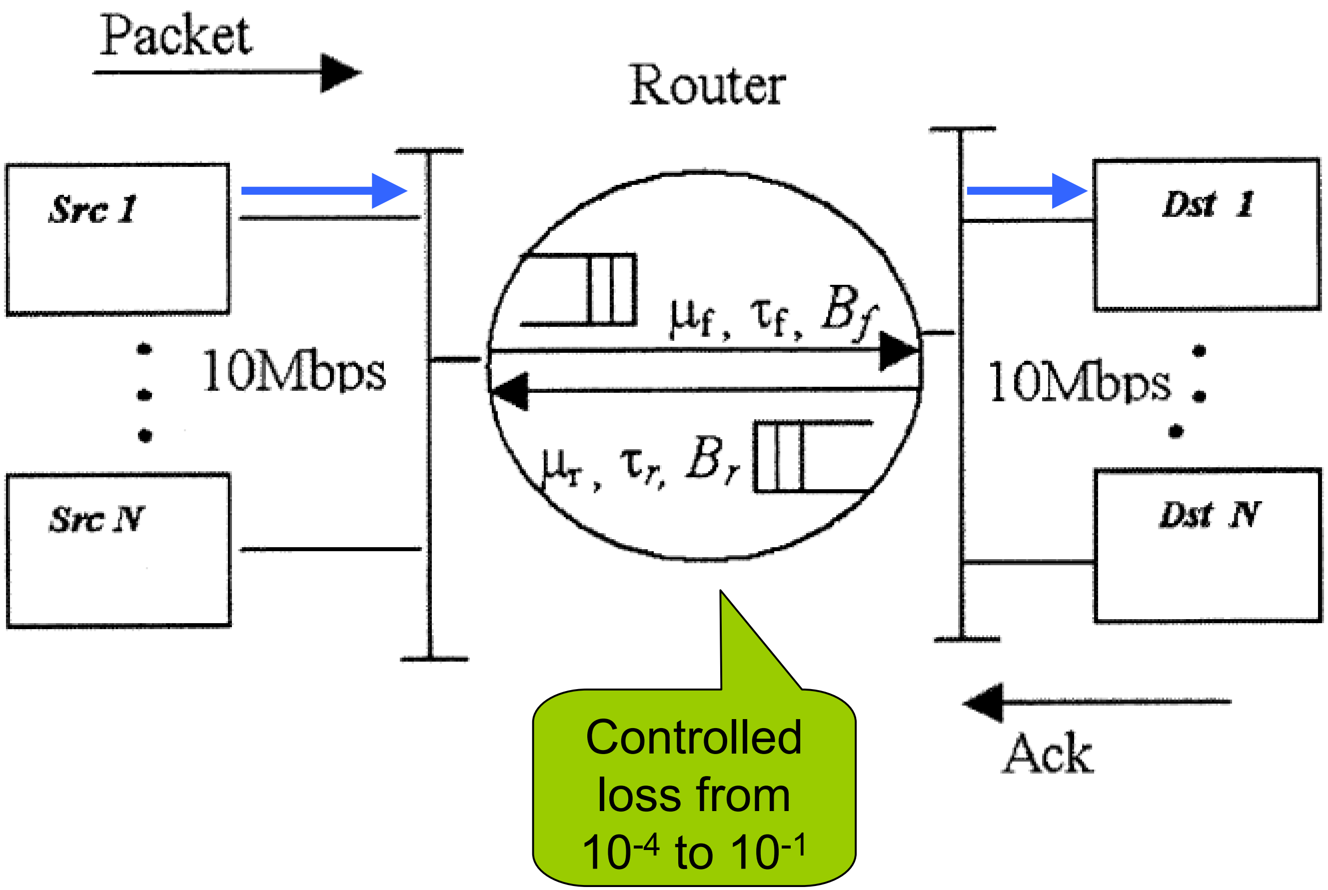


Simulation A: Distinguishing Packet Loss

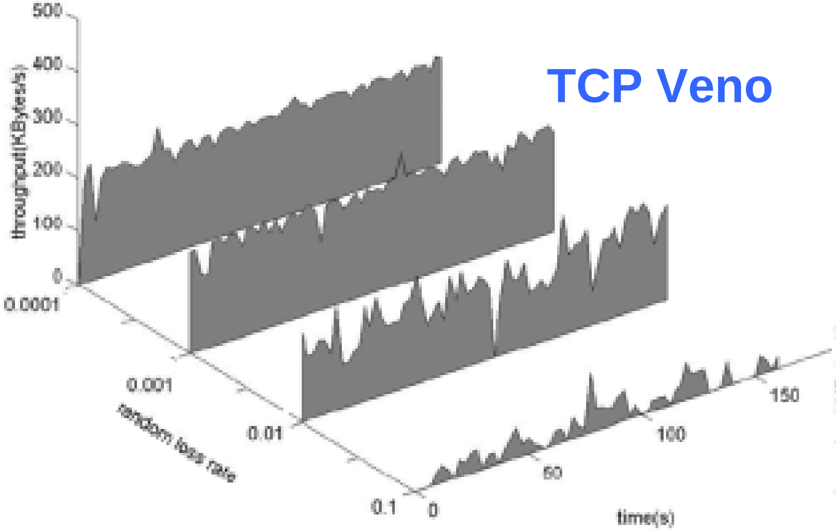
- Random loss detection accuracy: 83/84=99%
- Misdiagnosing rate of congestion: 2/12=17%

Veno TCP		No background traffic		Background traffic (UDP) with sending rate of 500kb/s		Background traffic (UDP) with sending rate of 1Mb/s	
Actual type of packet loss		Congestion	Random	Congestion	Random	Congestion	Random
First	#FF4/5	0	105	1	83	2	90
Run	#FF1/2	6	120	10	146	12	136
Second	#FF4/5	0	64	1	78	2	83
Run	#FF1/2	3	110	8	102	16	127
Third	#FF4/5	0	46	1	67	1	53
Run	#FF1/2	3	123	9	139	19	132

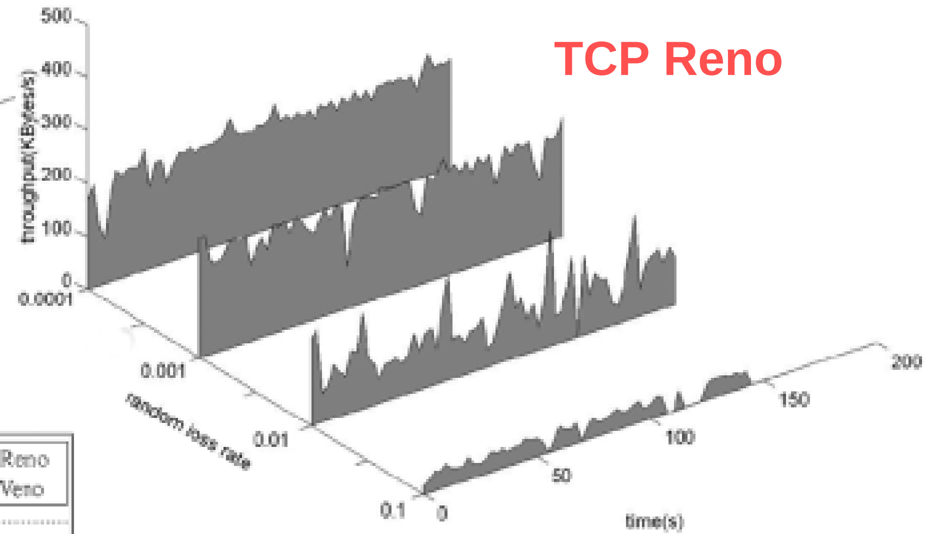
Simulation B: Single Connection



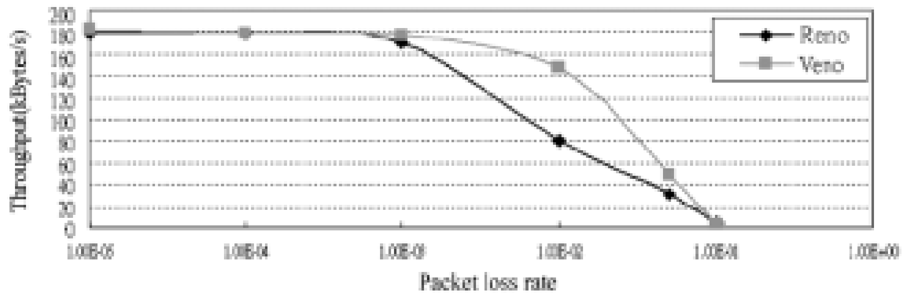
Simulation B: Single Connection



TCP Veno

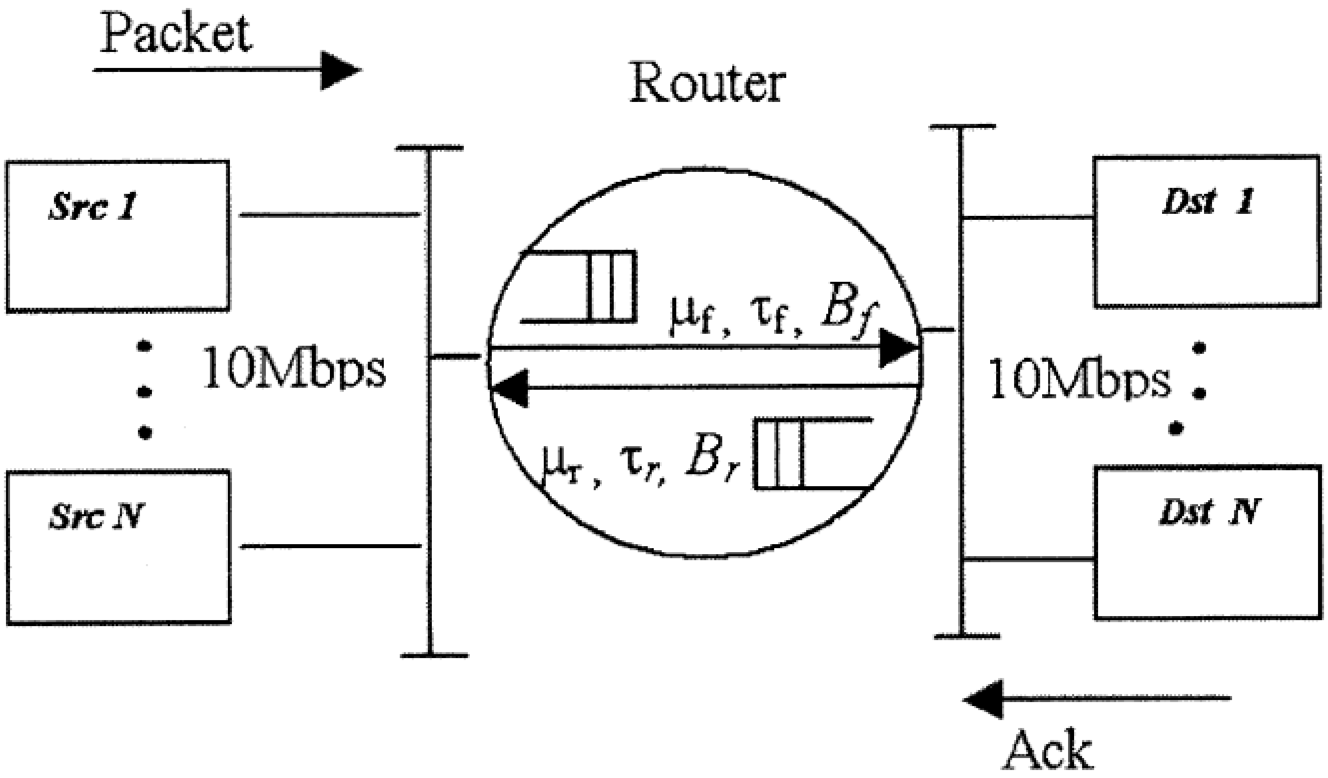


TCP Reno



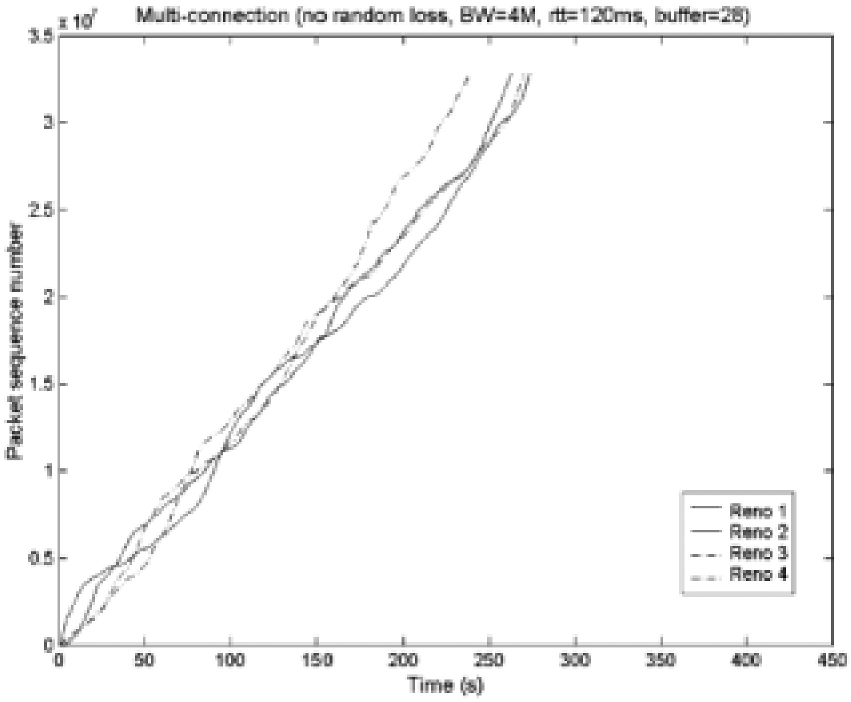
Simulation C: Co-existing Connections

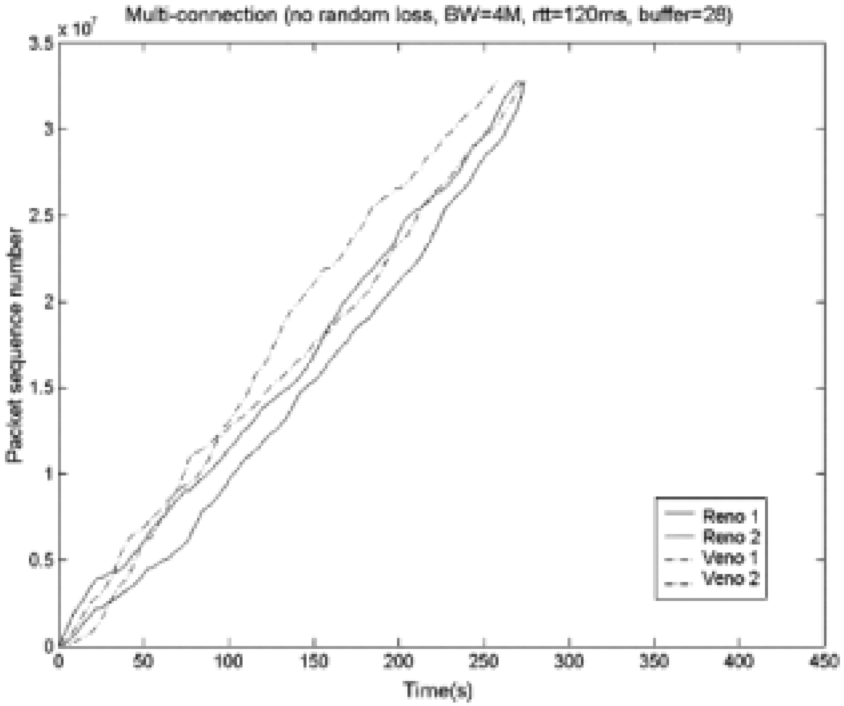
- 4R v.s. 2R2V



Simulation C: Co-existing Connections

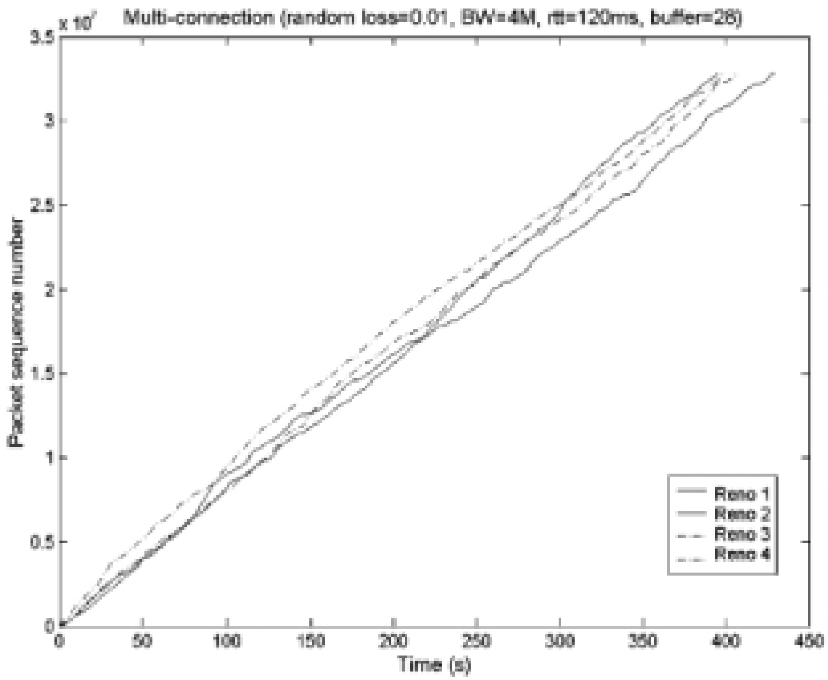
- Fair-share of R-V and V-V
- Same total throughput

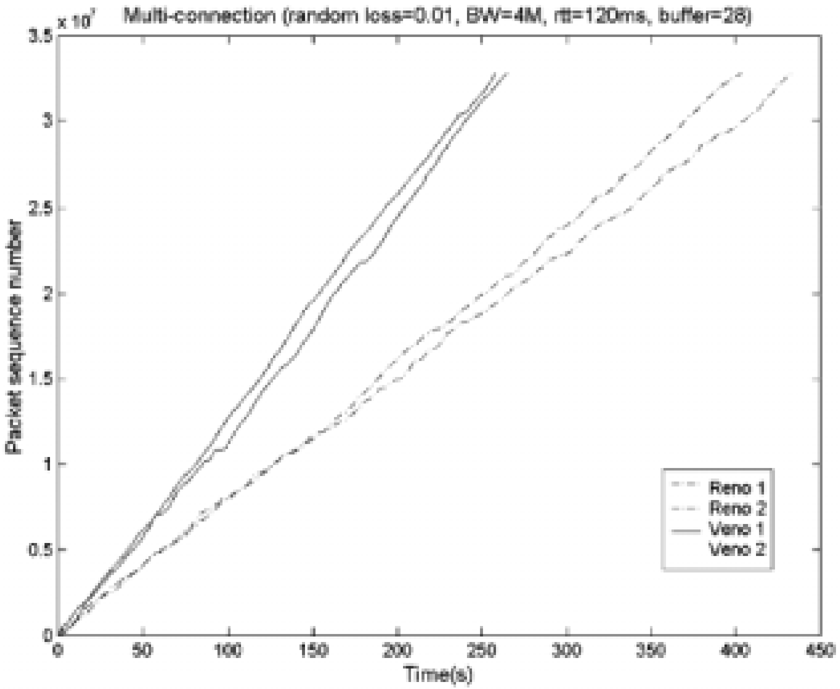




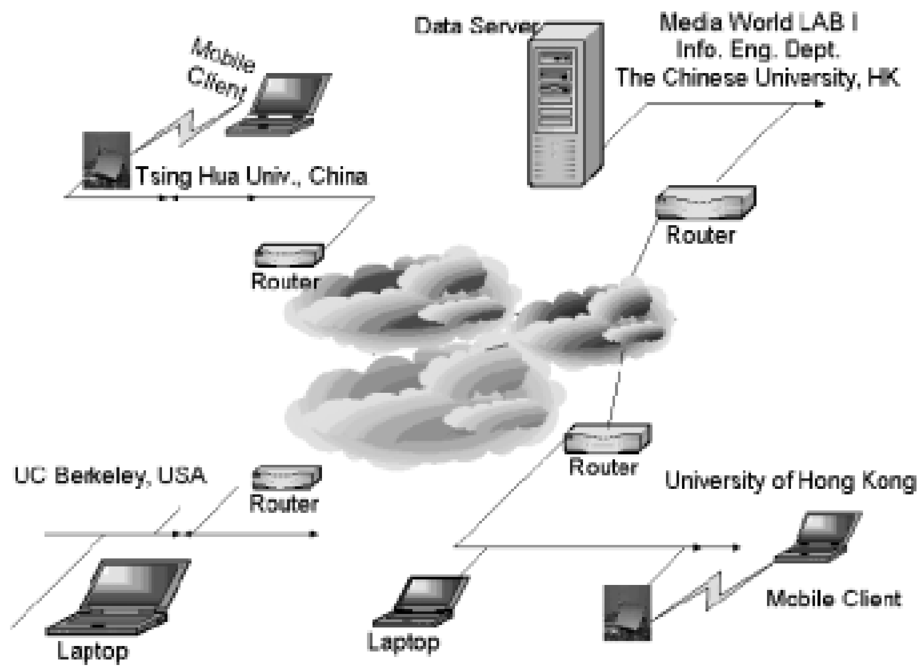
Simulation C: Co-existing Connections

- Non-interference between R and V
- Loss-tolerant of V (increased throughput)





Simulation D: Internet Measurements



Reno/Veno					
Time slots	10:00~11:00	11:00~12:00	12:00~13:00	14:00~15:00	15:00~16:00
Th(KB/s)	65.5/64.4	51.3/60.3	45.1/54.5/1.21	[illegible]1.5/58.7	70.2/73.9
# TO	8.1/6.5	7.6/5.8	8.9/5.3	9.5/5.8	6.5/6.4
# FF	19.9/11.5	21.6/10.5	24.6/18.5	28.3/18.7	14.5/14.0
# Retran. Pkts	185.8/127.5	166.7/103.9	170.1/137.3	156.6/137.2	98.5/97.1

Conclusions

- Veno is desirable for:
  - Deployability
  - Compatibility
  - Flexibility

- *What TCP Veno proposes is to refine Reno's AIMD evolution over heterogeneous networks by using the complete judgment of network state estimation – congestive state or non-congestive state, rather than merely depending on packet loss occurrence.*