

# TCP Veno: Solution to TCP over Wireless

**Franklin FU**

Presented by Franklin Fu

Asst Professor

School of Computer Engineering  
Nanyang Technological University  
Singapore

January 31, 2004, 5:00am Singapore

January 30, 11:00am Hawaii, USA

# Outline

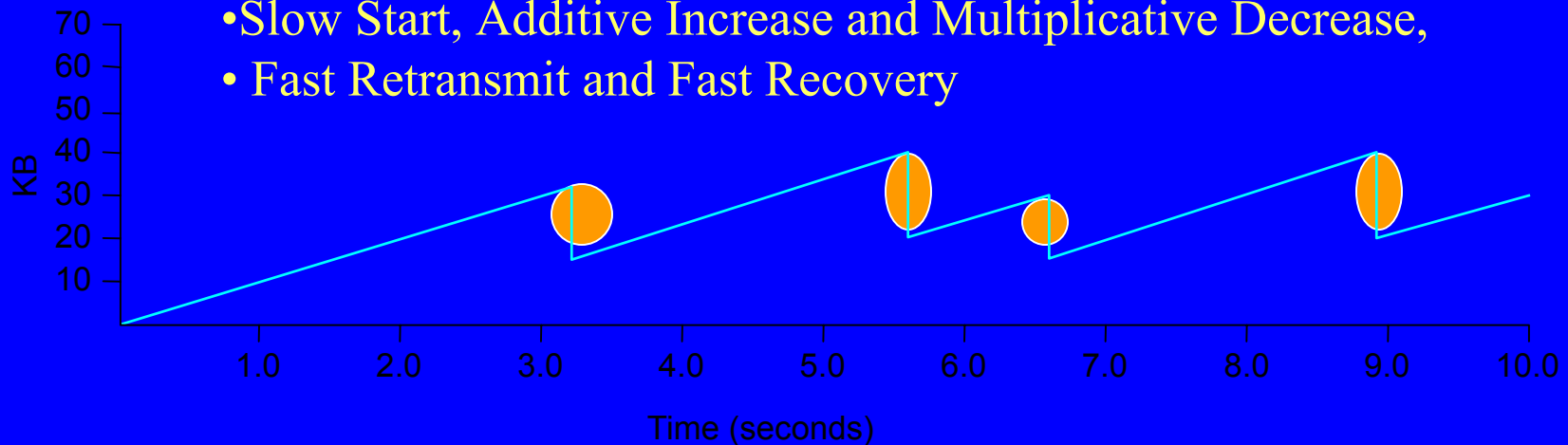
- What's TCP Veno ?
  - TCP Briefing
  - TCP Veno
- Common Issues Concerned
- Ongoing Work

# TCP Briefing

- TCP Reno 1988, 1990
  - V. Jacobson “Congestion Control and Avoidance” *ACM SIGCOMM 1988* (Real source codes available immediately)
- TCP Vegas 1994
  - L. S. Brakmo, L. L. Peterson “TCP Vegas: New Techniques for Congestion Detection and Avoidance” *ACM SIGCOMM 1994 & IEEE (JSAC) Journal of Selected Areas in Communications 1995* (Real source codes available immediately)
- TCP Veno 2001
  - C. P. Fu, S. C. Liew “TCP Veno: TCP Enhancement for Transmission Over Wireless Access Networks” *IEEE (JSAC) Journal of Selected Areas in Communications 2003* (Real source codes available immediately)

# TCP Reno --- SS AIMD FF

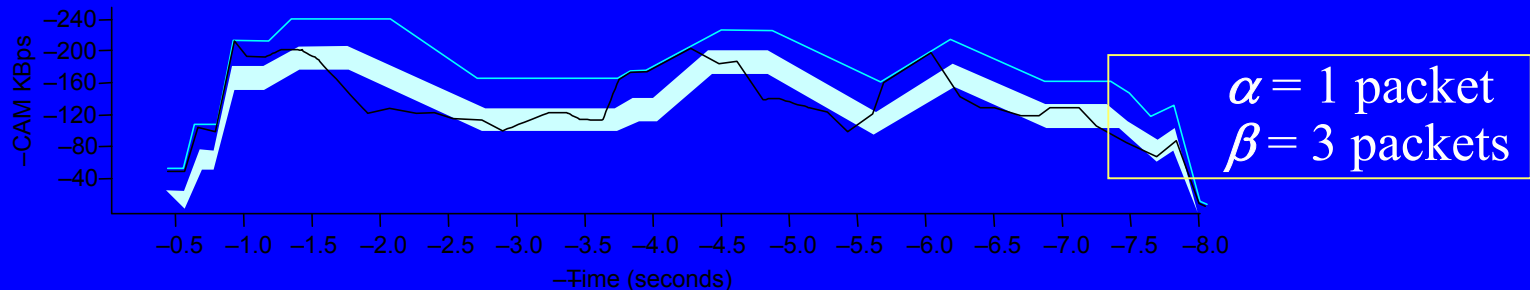
- Slow Start, Additive Increase and Multiplicative Decrease,
- Fast Retransmit and Fast Recovery



- 1988 Tahoe, 1990--Reno: sawtooth (fluctuation) behavior
  - aggressively increases window, resulting in **periodic congestion loss**
  - Reno estimates the equilibrium point of a connection using a rough mechanism (AIMD)
- Reno's variants: 1995 - 2000 (AIMD, **FF\***, SS)
  - NewReno, SACK, FACK, Rate-Halving, NetReno etc.

# TCP Vegas --- SS\* FF\* AIAD

AIAD: Additive Increase and Additive Decrease



## Proactive Congestion Control (1988-1995)

- Vegas “carefully” probes the equilibrium of a connection using the estimated # packets at bottleneck router
- Avoids occurrence of congestion loss – buffer overflowing
- Similar Algorithms:
  - DUAL, CARD, Tri-S, Packet-Pair, Santa Cruz

# Any Disadvantages?

- Shortcomings of Reno
  - Reno considers all packet losses to be indicators of network congestion
    - Cannot distinguish between random loss and congestion loss
  - Window is erroneously reduced in lossy environments
- Shortcomings of Vegas
  - Inability to distinguish between congestion loss and random loss,
    - Vegas “rigidly” treats all packet loss as random loss
  - Not compatible with large installed-based TCP Reno
  - Performance degrades substantially in asymmetric networks (i.e., ADSL)

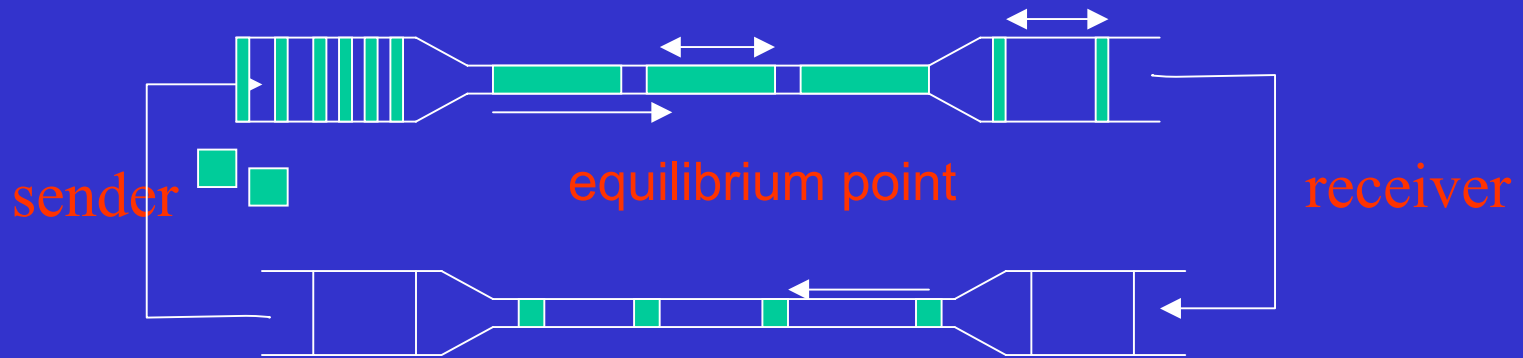
# Outline

- What's TCP Veno ?
  - TCP Briefing
  - TCP Veno
- Common Issues Concerned
- Ongoing Work

# TCP Veno

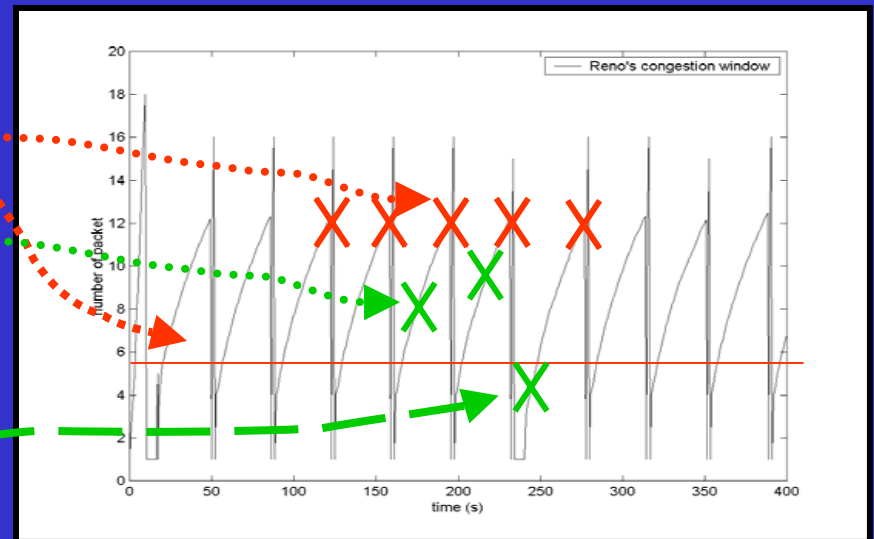
- Combines elements from two **opposing** TCP camps:
  - 1) Reno which uses reactive congestion control; and
  - 2) Vegas which uses proactive congestion control.
- Specifically, integrate the idea of *Congestion Detection* of Vegas TCP into Reno TCP in order to
  - Distinguish connection evolution in congestive or non-congestive state, so that window size can be adjusted rationally
  - Extend Additive Increase phase to better make use of available bandwidth and reduces occurrences of congestion loss

# Congestion Loss / Random Loss vs Congestive State / Non-congestive State



$$BDP = \text{Bandwidth} * RTT$$

Congestion loss  
Random loss or  
Congestion loss ?  
Random loss



# TCP Veno Mechanism (1)

## ----- Refining Reno's MD

- If loss occurs when not in congestive state, declare random loss; otherwise, declare congestion loss
- Algorithms:

when packet loss is detected by fast retransmit:

```
if (DIFF*BaseRTT <  $\beta$ ) ssthresh =  $\text{cwnd}_{\text{loss}} * (4/5)$ ;  
    //where DIFF=(cwnd/BaseRTT - cwnd/RTT )*BaseRTT  
    //random loss ( due to bit errors ) is most likely to have occurred
```

```
else if ssthresh =  $\text{cwnd}_{\text{loss}} / 2$  ;  
    // congestive state is most likely to have occurred,  
    //even there occurs random loss at this time
```

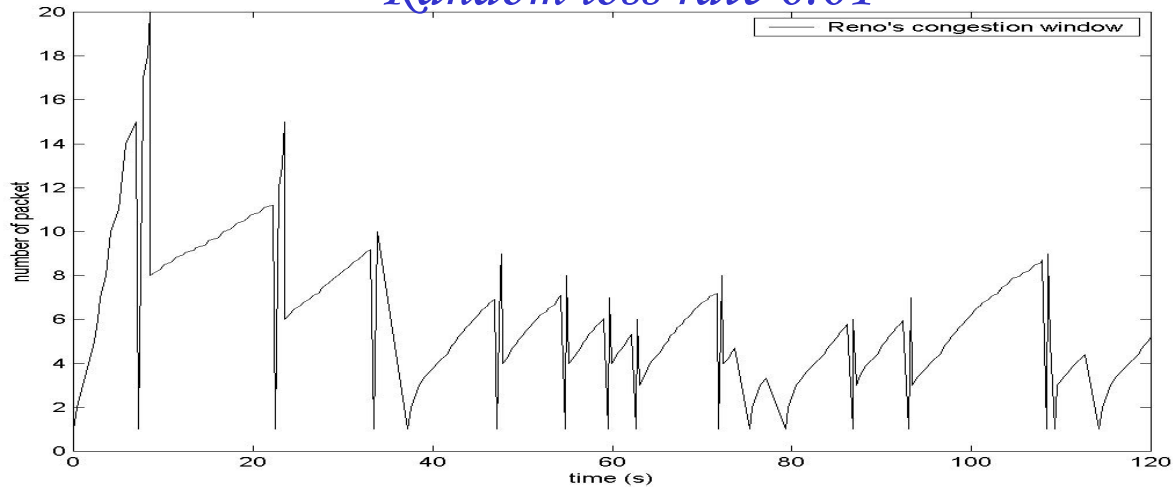
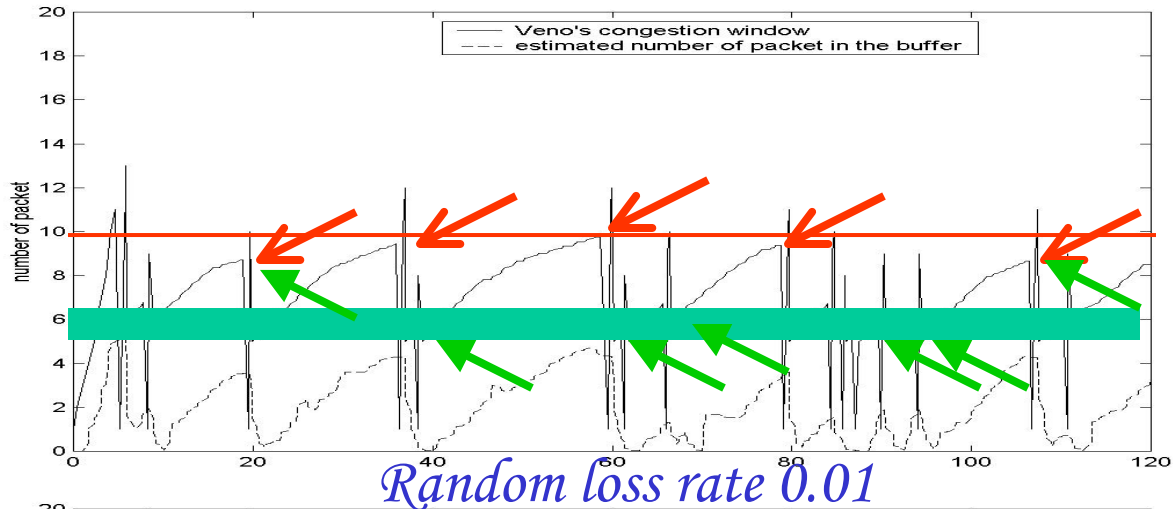
when packet loss is detected by retransmit-timeout timer:

*ssthresh* is set to half of the current window ;

slow start is performed; // performs the same action as in Reno

# TCP Evolution Snapshot in Real Network

Veno  
80%  
Over  
Reno  
Reno



# TCP Veno Mechanism (2)

## ----- Refining Reno's AI

- Algorithms:

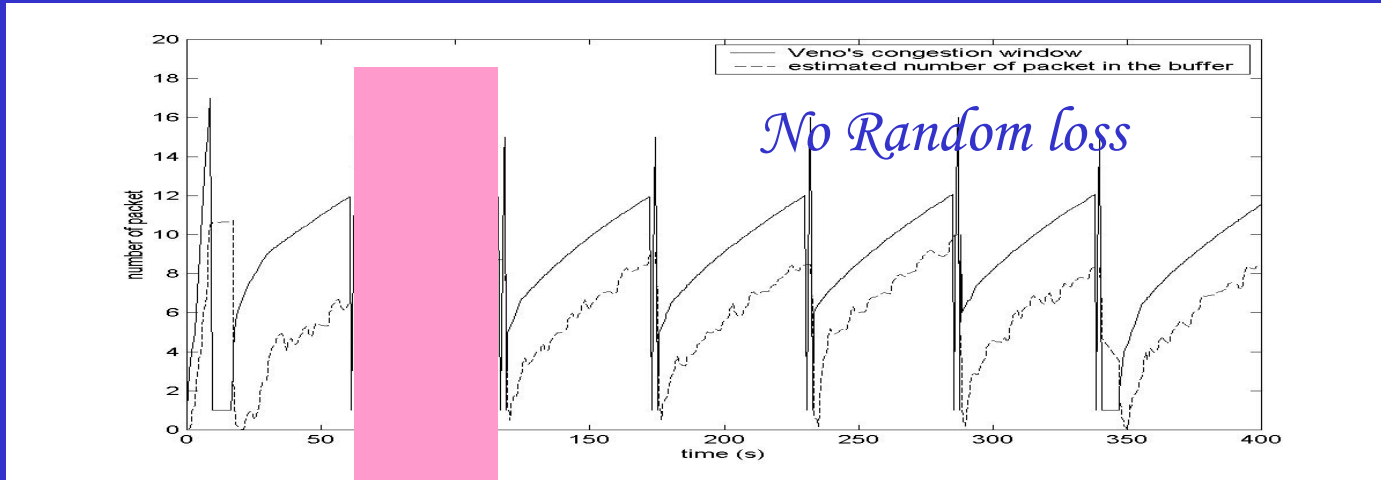
during the **additive increase (AI)** period,

```
if ( DIFF*BaseRTT ≤ β )           // available bandwidth under-utilized
    cwnd=cwnd+1/cwnd              // for every new ack received
else if (DIFF*BaseRTT > β )      // available bandwidth fully utilized
    cwnd=cwnd+1/cwnd;            //for every other new ack received
```

- AI dominates most part of the connection evolution
- Try to stay at the ideal transmission rate as long as possible.

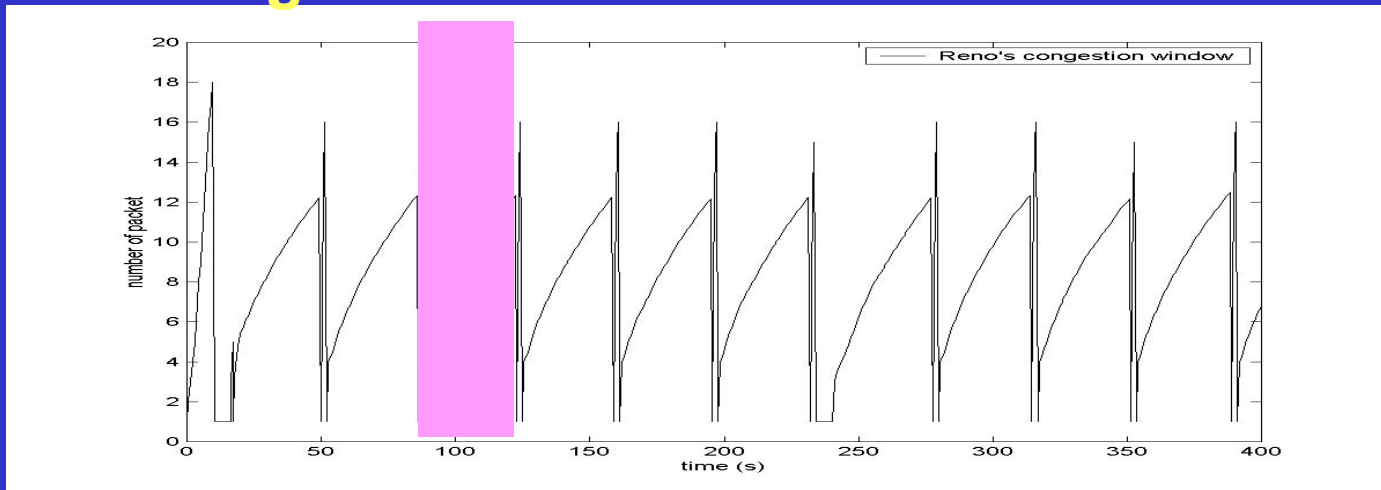
# Cont'd

Veno



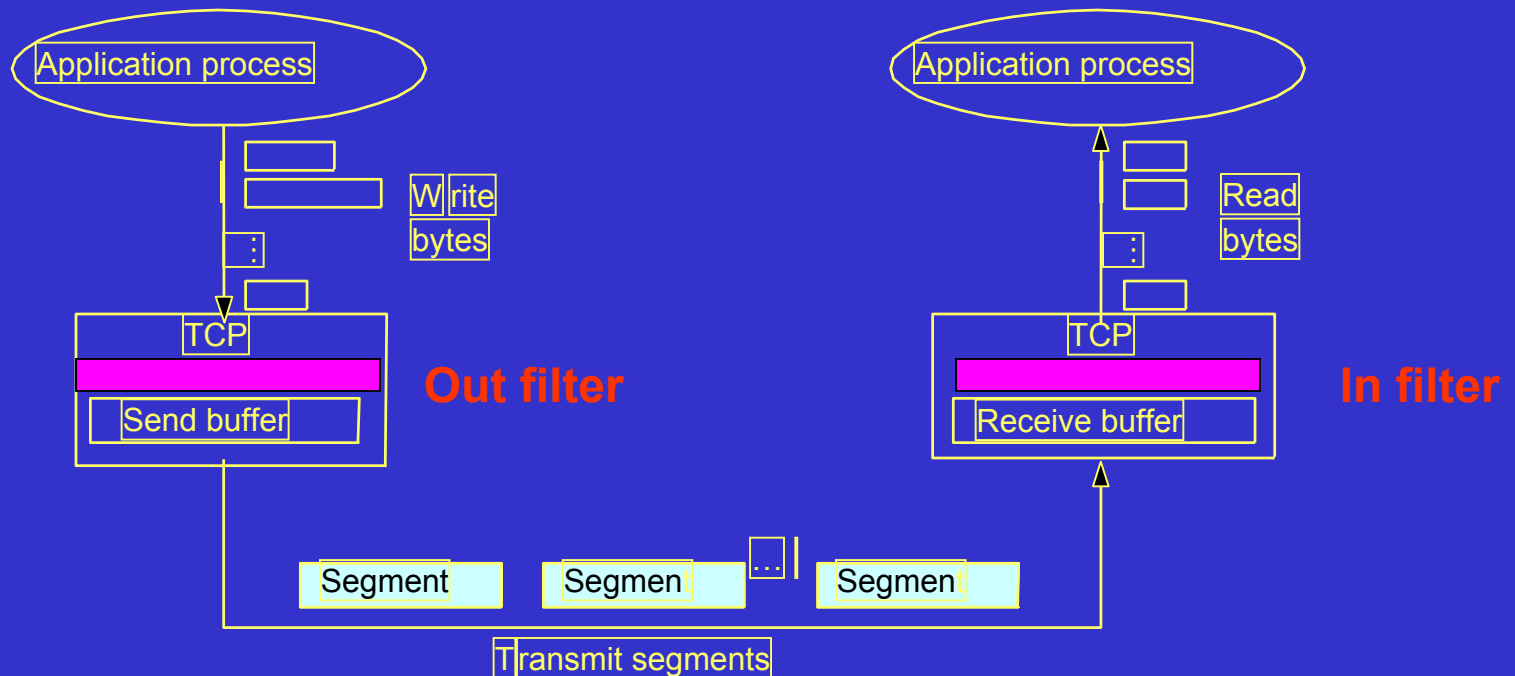
40% congestion loss is reduced

Reno

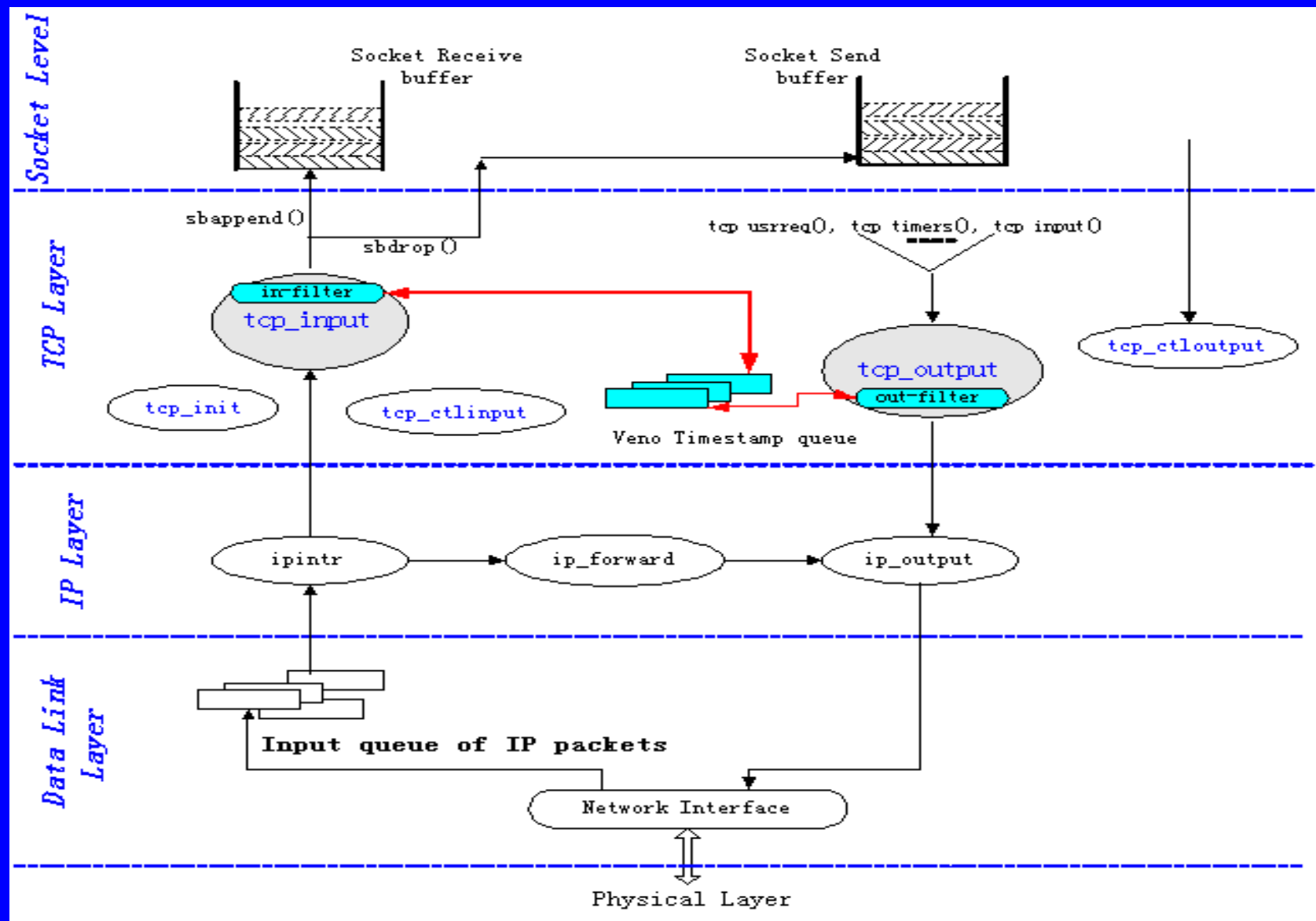


# How to Implement TCP Veno?

- Logical concept



# More Details



# Outline

- What's TCP Veno ?
  - TCP Briefing
  - TCP Veno

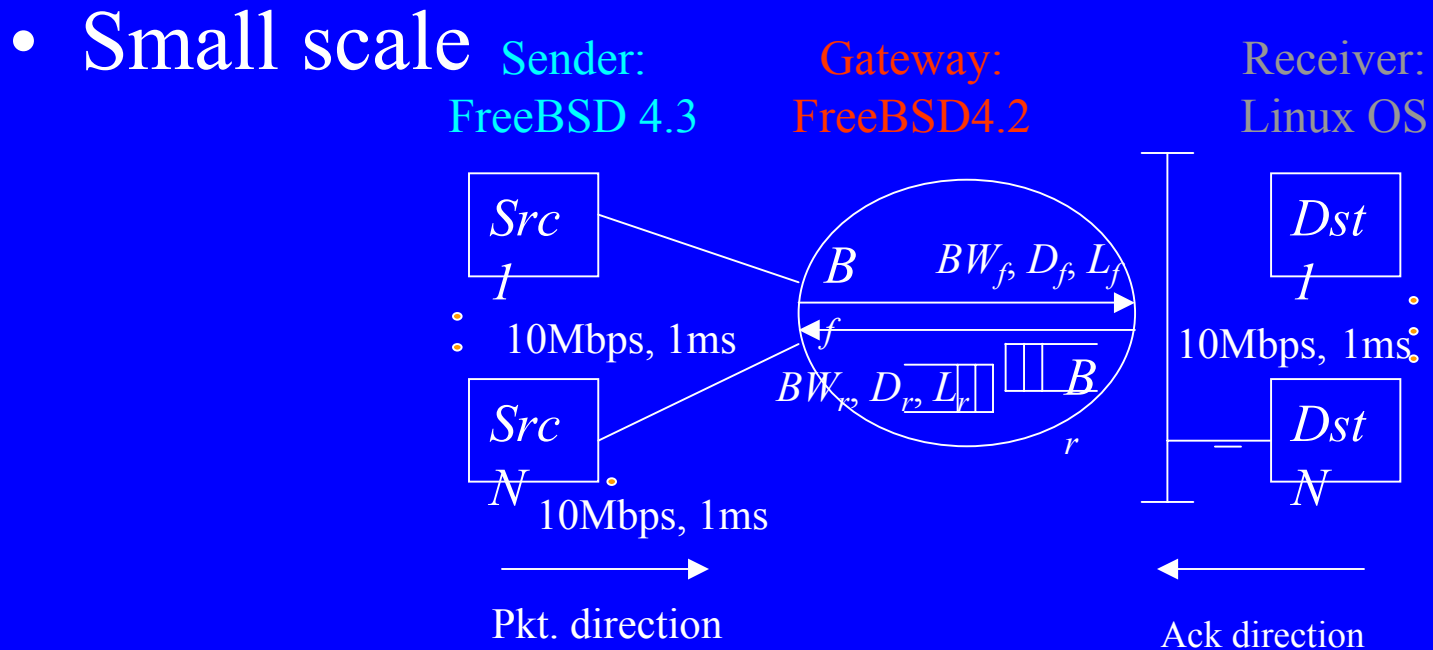
Common Issues Concerned

Ongoing Work

# How to Evaluate TCP?

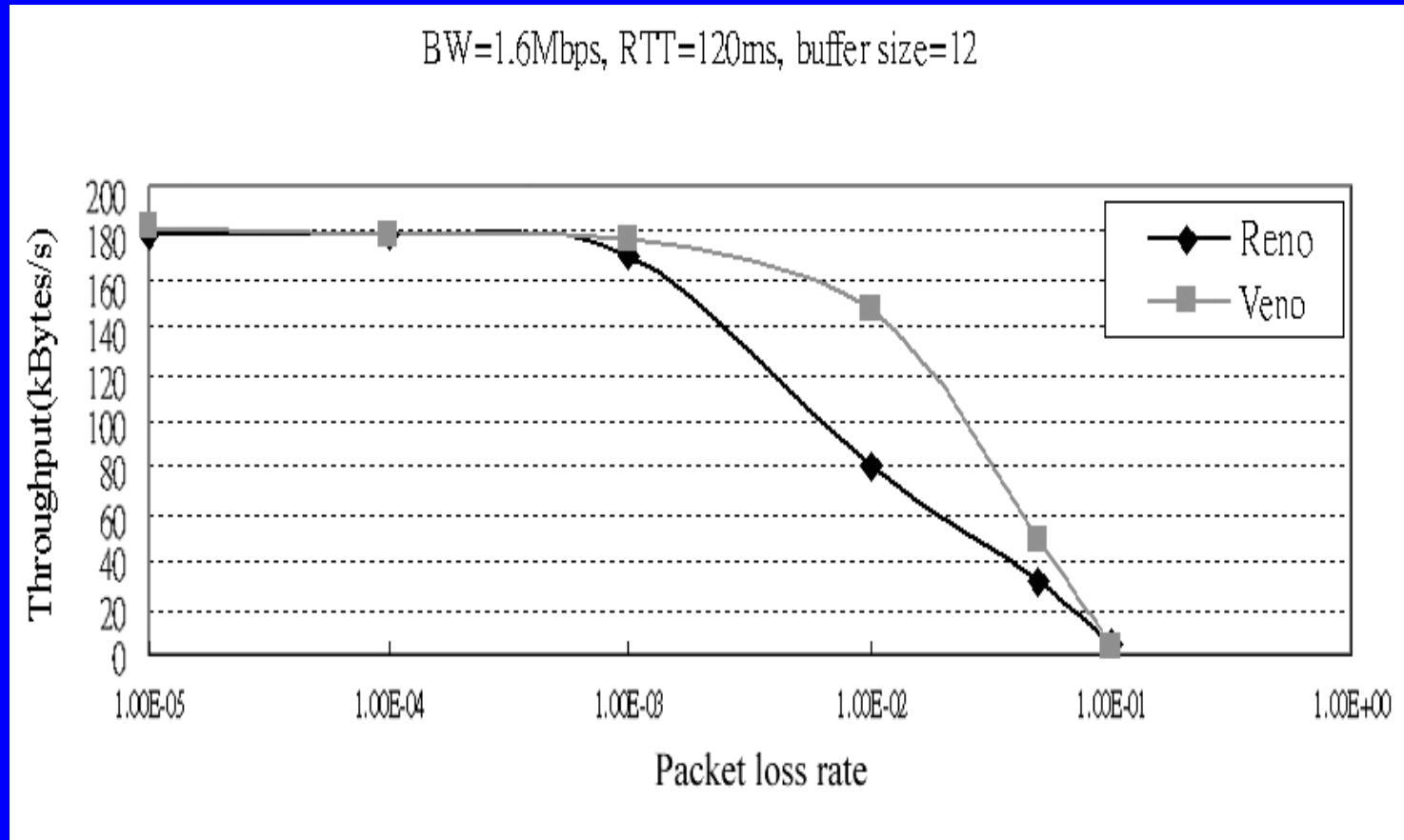
- Throughput (for a single connection)
  - Higher, better?
  - Need evaluation in one scenario or in different scenarios ?  
*Flexibility*
- Compatibility (for multi-connections)
  - Within own connections: *Fairness definition* =  $(\sum_{i=1}^n b_i)^2 / n * (\sum_{i=1}^n b_i^2)$
  - With installed-based legacy TCP ( Higher throughput from better efficient utilization rather than grabbing bandwidth from other TCP connections)
- Robustness
- Deployability (practical consideration)

# Experiments

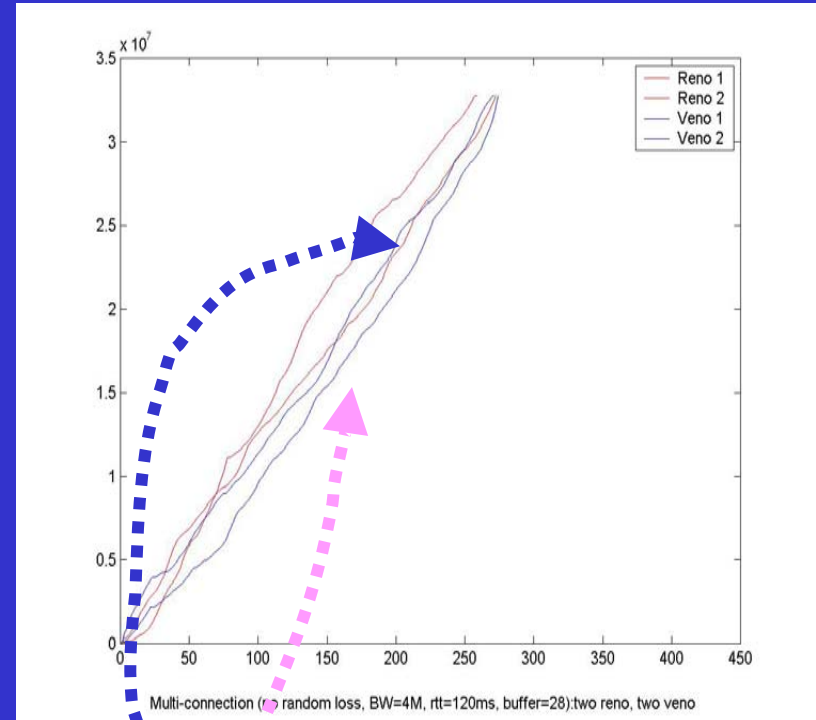
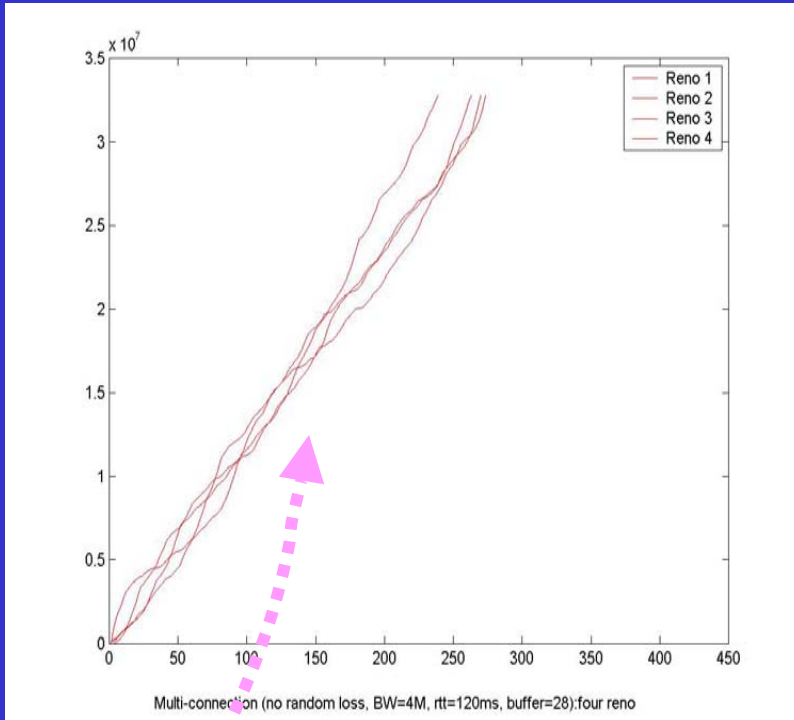


- Large scale --- live Internet

# Throughput



# Compatibility & Throughput

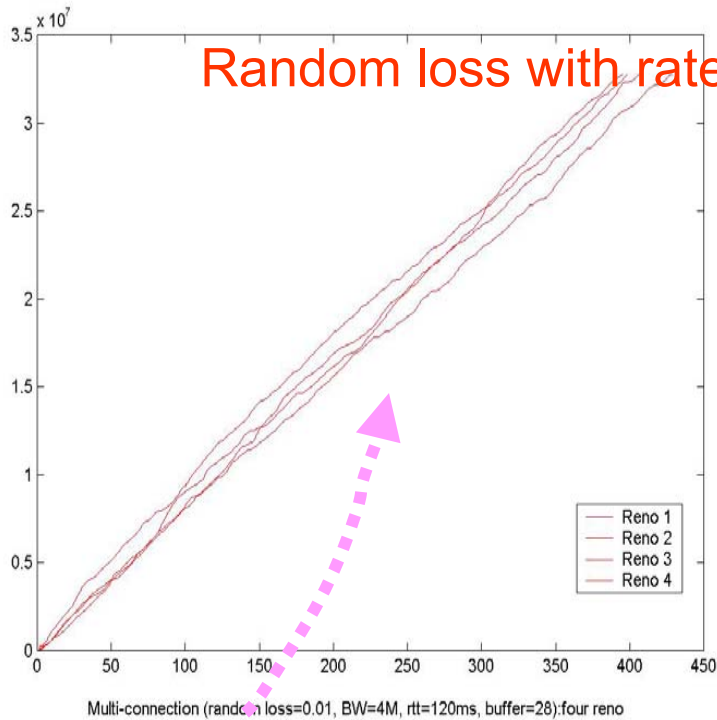


- **Four Reno** connections

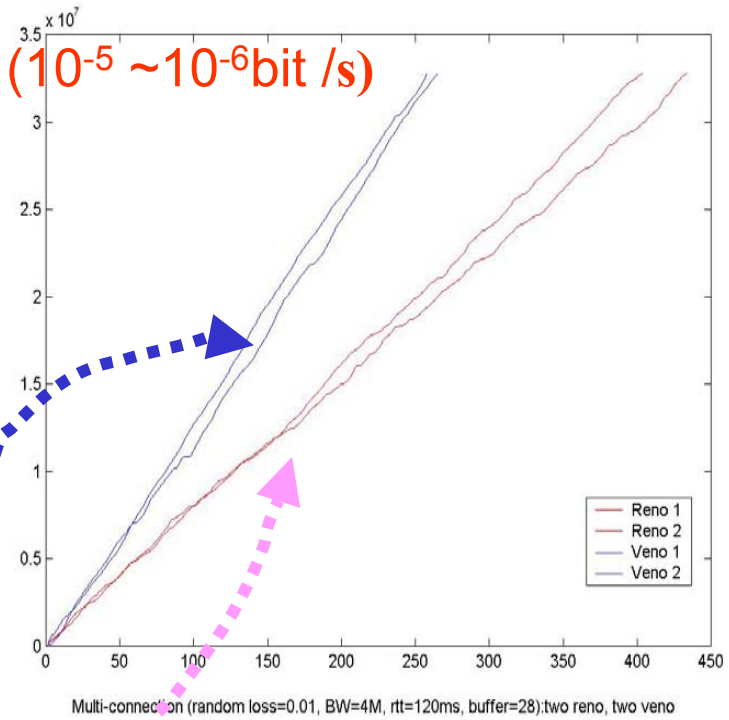
- **Two Reno** connections
- **Two Veno** connections

- **No random loss** in this situation
- **Fairness/Compatible?**

# Cont'd



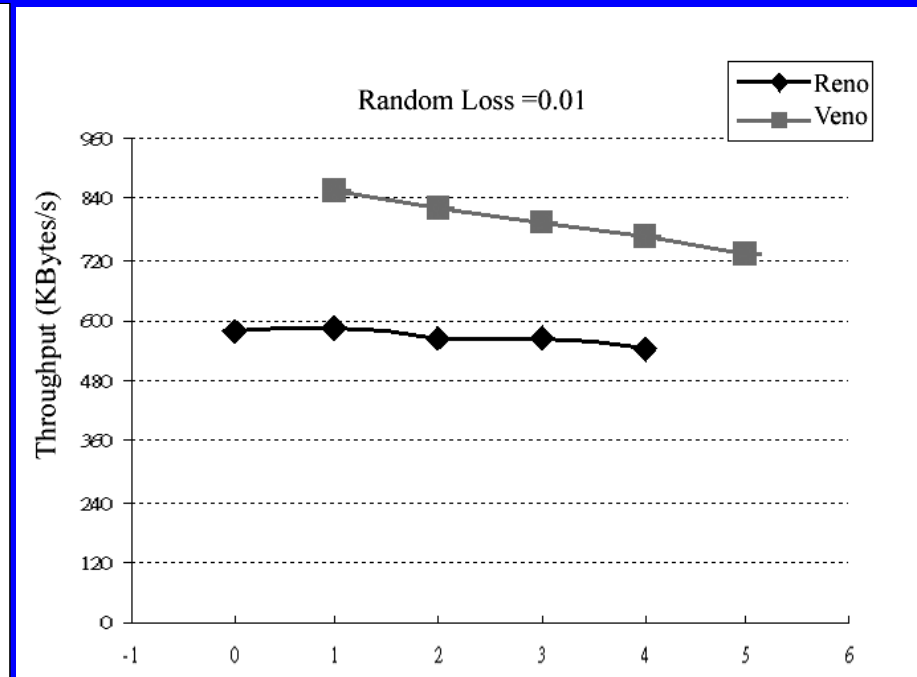
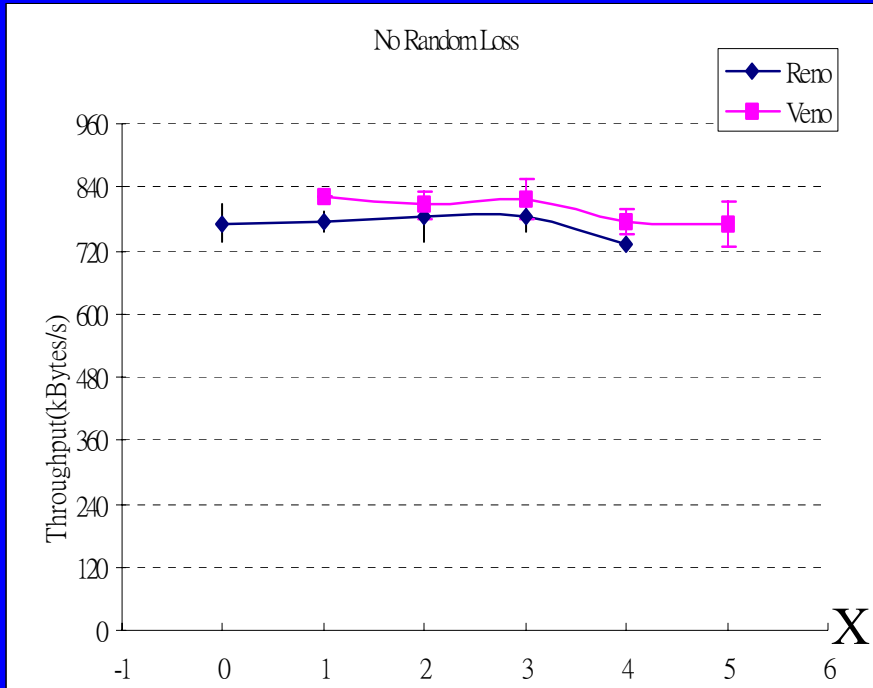
• **Four Reno** connections



• **Two Reno** connections  
• **Two Veno** connections

80% improvement over Reno for single connection with loss rate=0.01

# Cont'd



Total connections is 5 with  $X$  Veno connections and  $(5-X)$  Reno connections

- 1) Veno achieves higher throughput
- 2) Veno doesnot steal bandwidth from its counterpart since curve is horizontal
- 3) Fair among Veno's connections since standard deviation is small

# Robustness

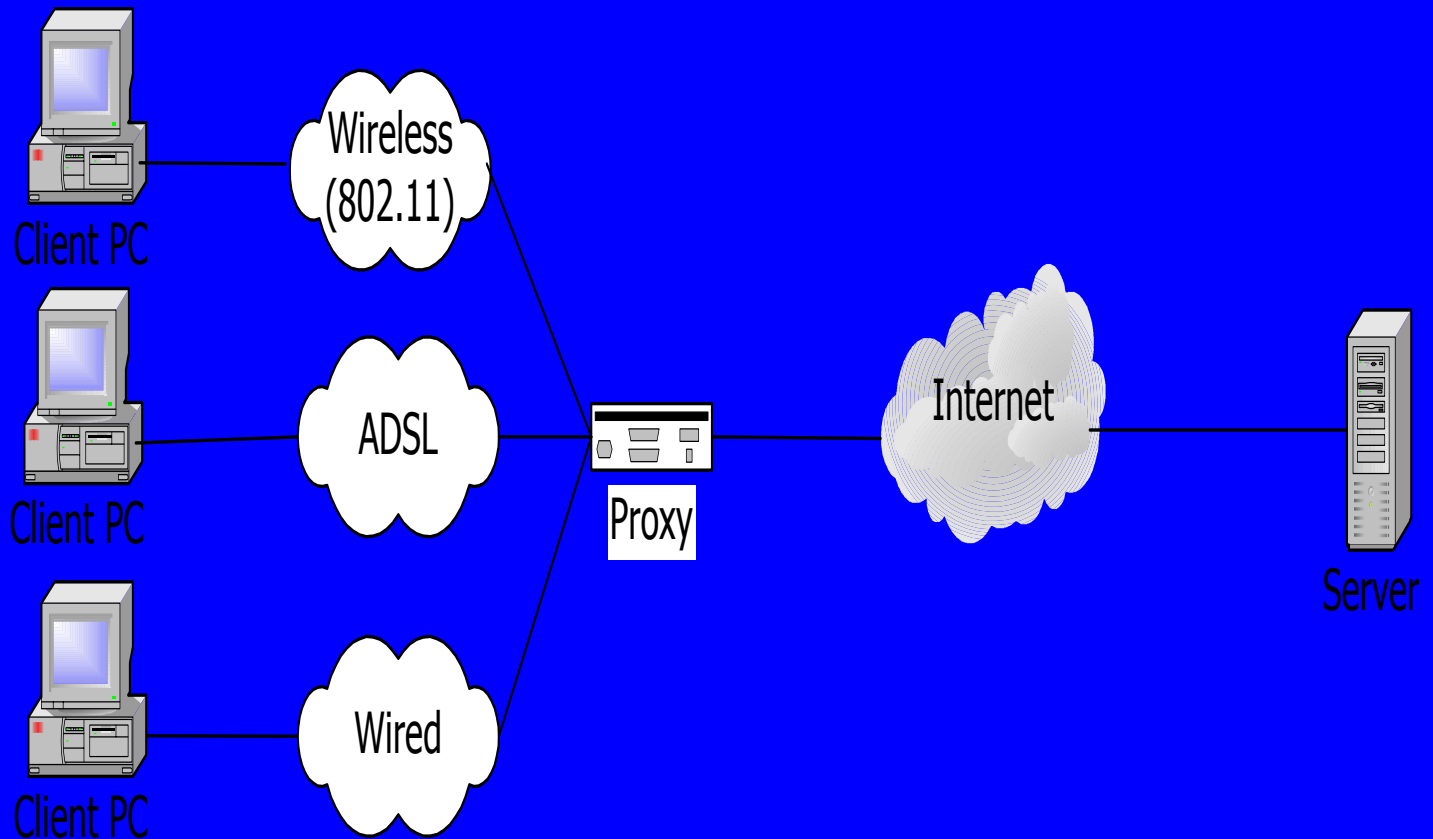
- Robustness
  - Repeat 1980's Internet collapse ?
  - Veno comply with the principle of “*Protocol should be designed defensively*”, and “*be liberal in what you accept but conservative in what you believe*”.
  - Dah-Ming Chiu, Raj Jain, “ Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks,” Journal of Computer Networks and ISDN, Vol. 17, No. 1, June 1989
  - T. Anderson, S. Shenker, I. Stoica, and D. Wetherall, "Design Guidelines for Robust Internet Protocols", HotNets-I, October 2002.

# Deployability ?

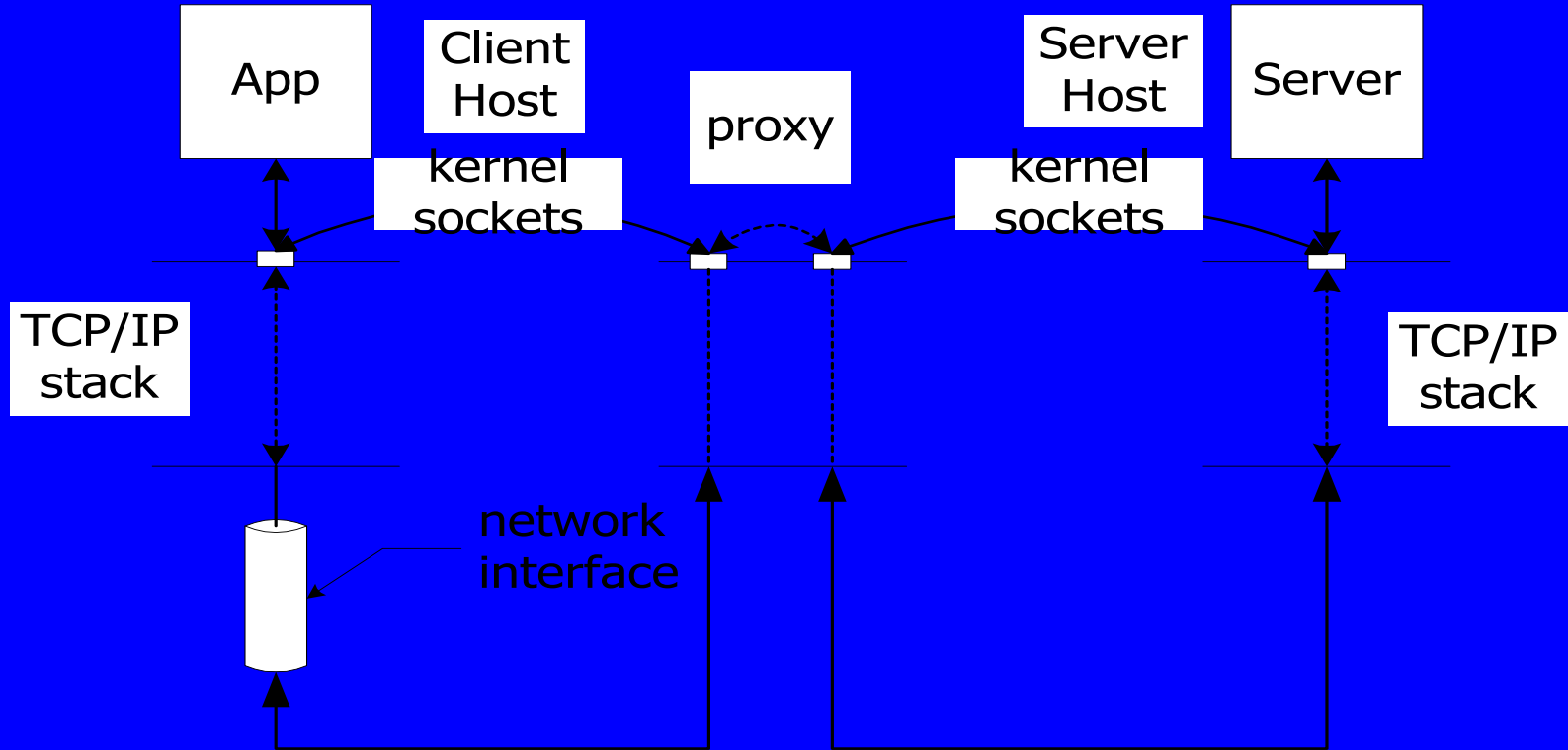
- One side modification?
- Two sides modification?
- Need support from Intermediate node?
- Any more?

Which do you prefer?

# Deployability



# Cont'd



# Summary of TCP Veno

		VENO	Vegas	Veno
Flexibility in different networks	Normal[1]	general	general	better than Reno
	Wireless	very bad	good	good
	Asymmetric	general	very bad	better than Reno
Compatibility with	Reno	good	bad	good
	Itself	good	good	good
Robustness		good	general	good
Deployability		N/A	sending side	sending side
Congestion control with	Proactive	no	yes	yes
	Reactive	yes	no	yes

# Outline

- What's TCP Veno ?
  - TCP Briefing
  - TCP Veno

Common Issues Concerned

Ongoing Work

# Ongoing Work

- Extend Veno TCP to Multicast in Wired/wireless Networks
- Dynamics Study of Veno TCP
  - Synchronization in TCP Reno
- Work with ECN/RED?
- Further theory Analysis ?
- .....

# Acknowledgements

- Team Members: Miss Gigi Chung and Mr. Lui Hung Ngai, Wang wei, Chang chun lei, C.H. Foh
- Prof. Victor Li – General Chairman of INFOCOM 2004
- Prof. Li Xing and Dr. Wang Ji Long at **Tsinghua Univ**, China
- Dr Vern Paxson at ICIRI, Berkeley.
- Prof. Lee Bu Sung at Singapore, Prof. Lawrence Yeung at HKU,
- And ....

THANK YOU!

Q&A