ENSC 835/CMPT 885: HIGH-PERFORMANCE NETWORKS

Final Project Presentation -- Spring 2002

Evaluation of Different TCP Congestion Control Algorithms

Hilary Zhang

Zhengbing Bian

{hzhang|zbian}@cs.sfu.ca

Outline

- Introduction
- Our Approach
- Implementation Detail
- Results Discussion
- Reference

Why is Today's Topic Important

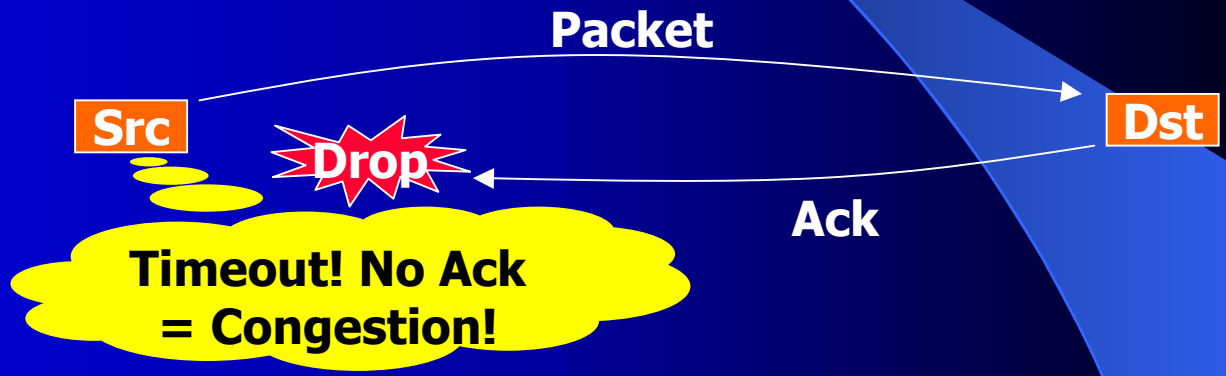
- **The algorithm for TCP congestion control is the main reason we can use the Internet successfully today.**
- **Without TCP congestion control, the Internet could have become *history* a long time ago.**

Main Categories of Congestion Control Approach

- Router-Centric vs. Host-Centric
  - Router makes decision and informs the host
  - Host adjust sending rate
- Reservation-Based vs. Feedback-Based
  - Fixed capacity connection
  - Adjust rate according to feedback
- Window-Based vs. Rate-Based
  - Advertise window
  - Reservation of a fixed bandwidth

TCP Congestion Control

- Host-centric
- Feedback-Based



- Window-Based
  - *congestion window* – window is smaller when congestion is larger and vice versa

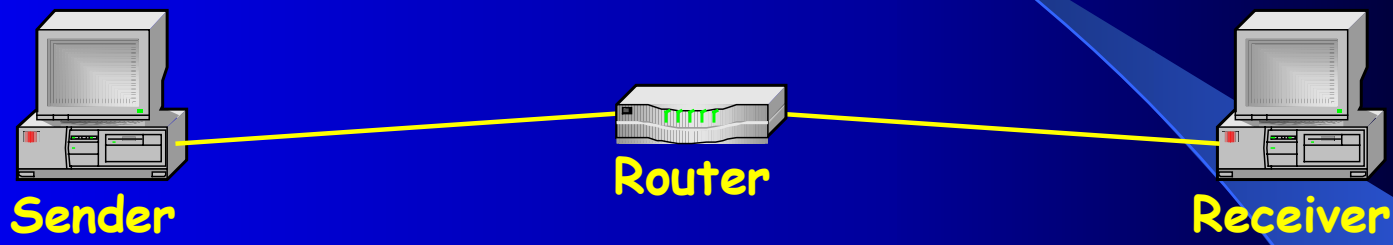
Review of TCP Congestion Control

- Slow Start, AIMD, Fast Retransmit, Fast Recovery
- Tahoe, Reno, NewReno, Sack, Vegas
- Evaluation measures
  - Effective resource utilization
  - Fair resource allocation

Our Approach

- Evaluate Reno, SACK, Vegas
- Effective Recourse Utilization
- Fairness between different delay links
- Competition between different versions TCP
- Effect of different Queuing algorithms

Effective Recourse Utilization(1/2)



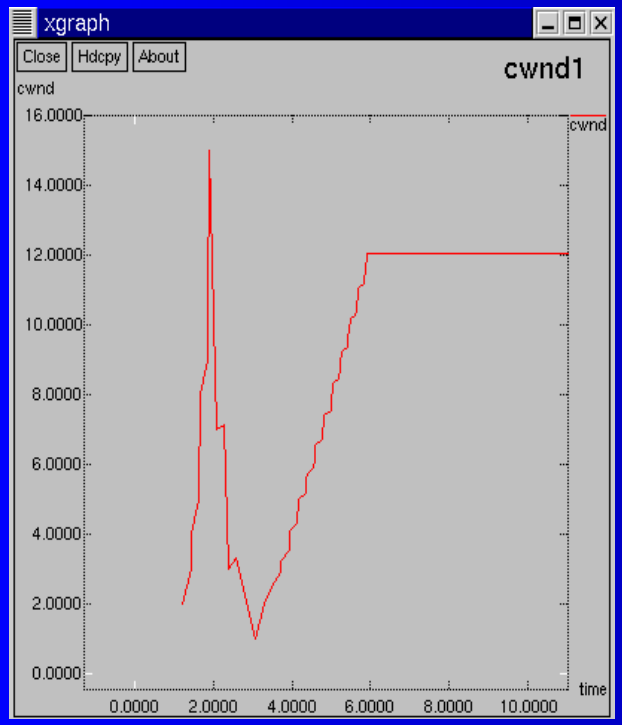
		Reno	SACK	Vegas
Bandwidth Utilization	1% loss	88.3%	90.2%	98.5%
	5% loss	34.5%	42.7%	75.0%

Effective Recourse Utilization(2/2)

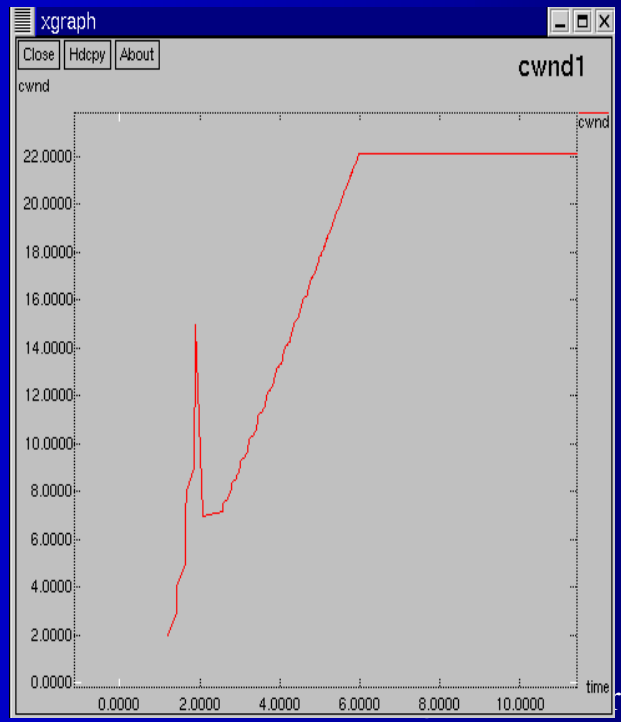
Congestion Window size variation

Four lost in one window

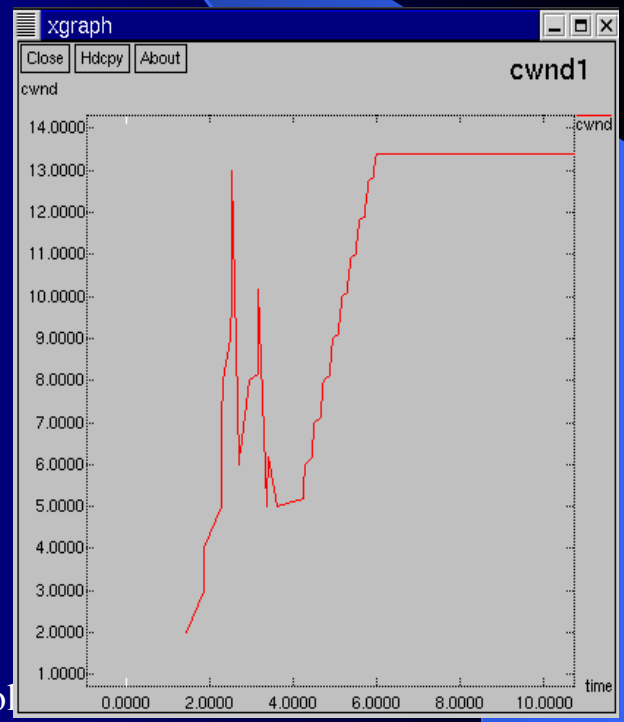
Reno



SACK

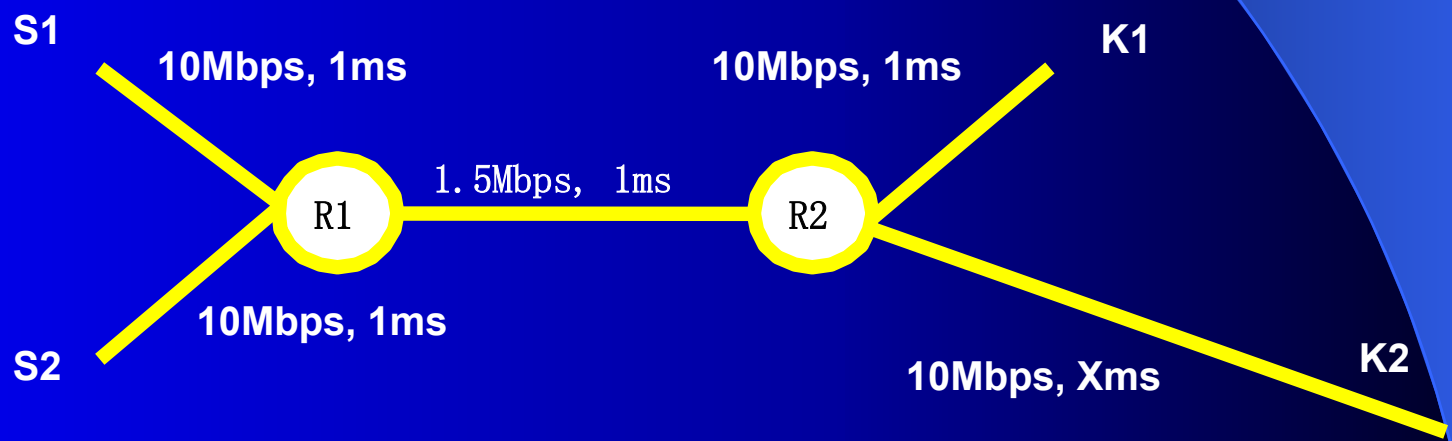


Vegas



Fairness Between Different Delay Links

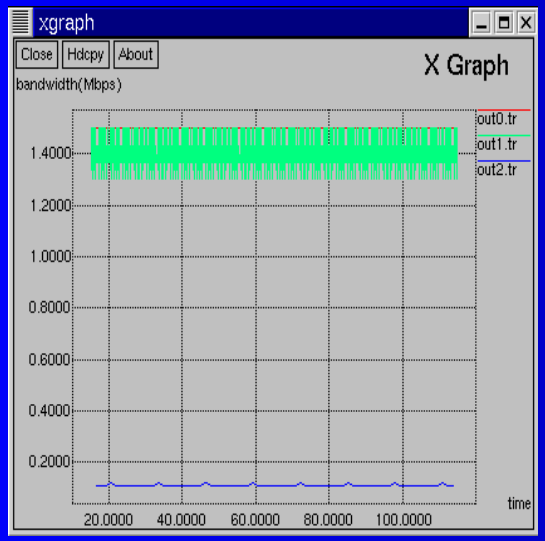
Simulation Topology



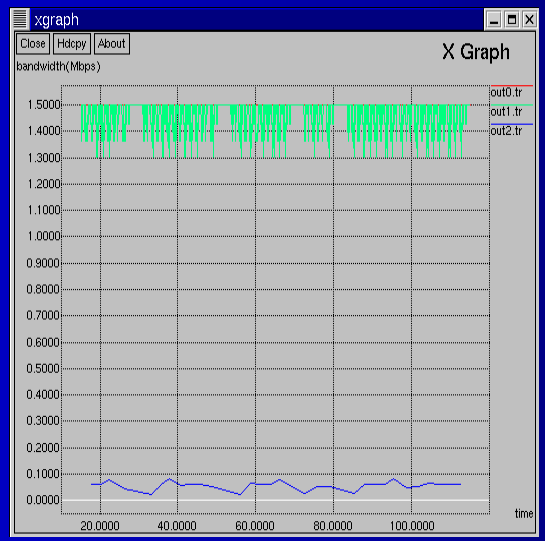
Bias on Long Delay Link

Bandwidth Occupation
(the same buffer size for all the three)

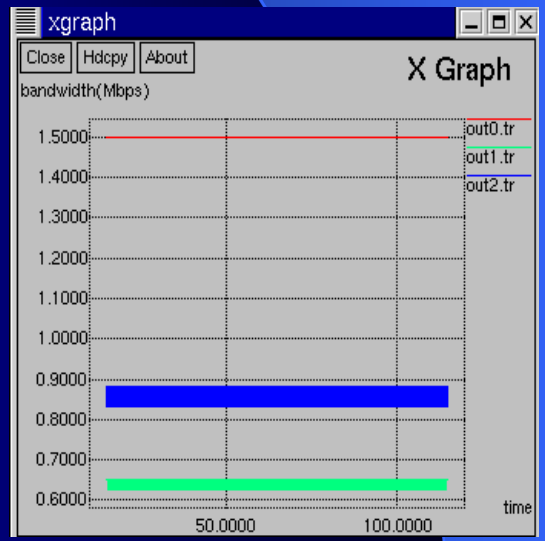
Reno



SACK



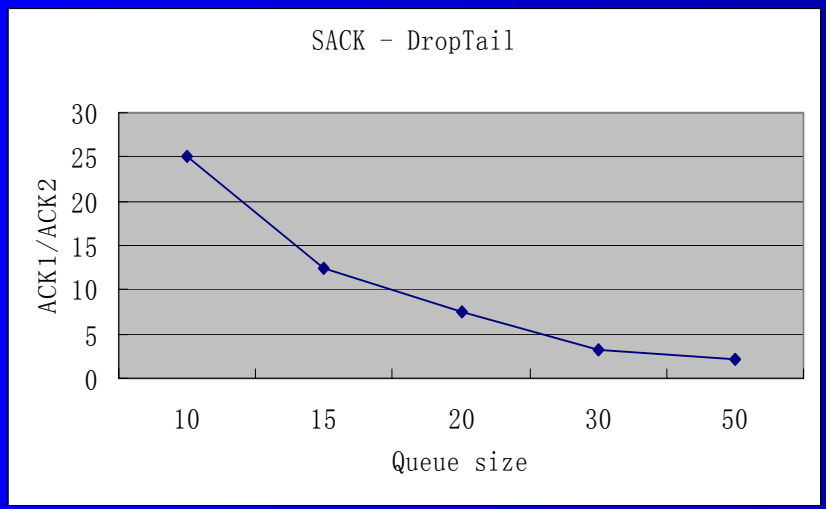
Vegas

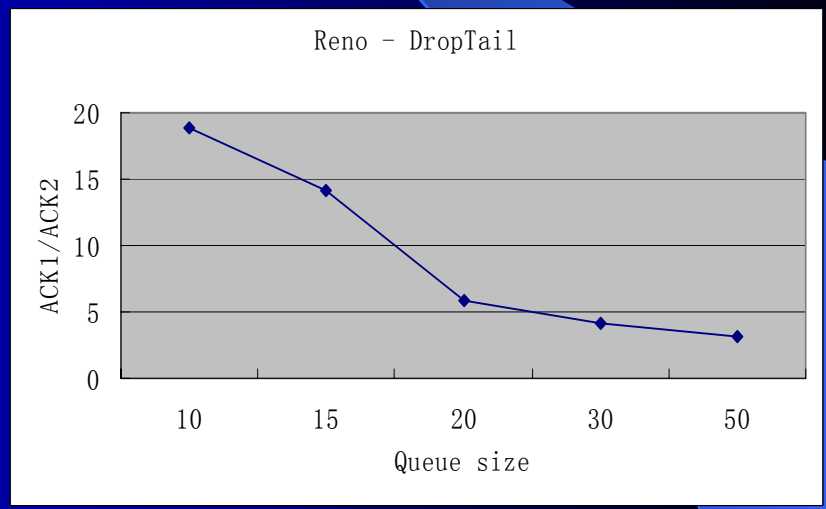


 Buffer size=10,Long delay=58ms

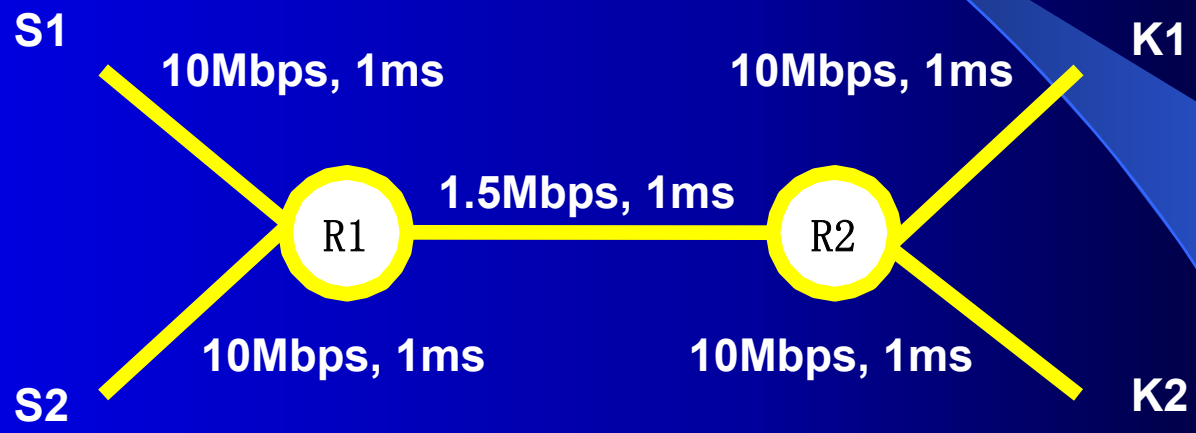
Queue Algorithm Effect

Fairness over Buffer size Changes





Competition Between Different Versions TCP



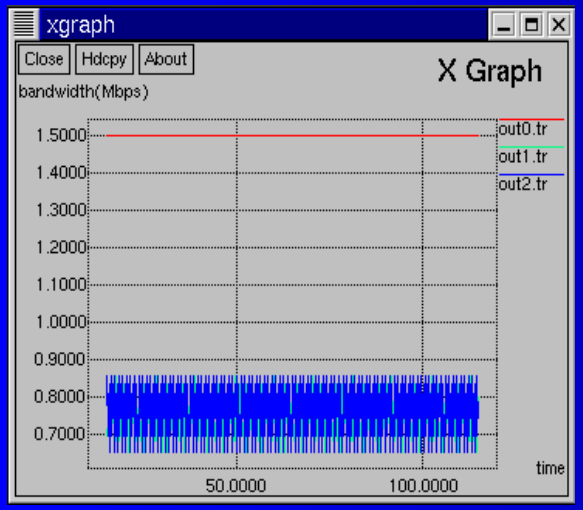
S1	S2
Reno	SACK
Reno	Vegas
SACK	Vegas

Buffer size=15

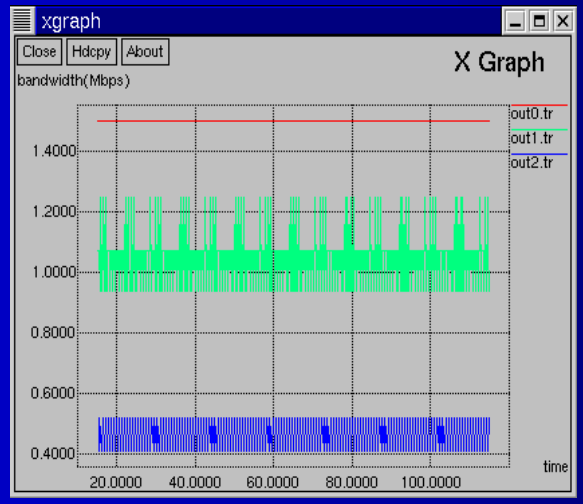
Simulation Results

Bandwidth Occupancy

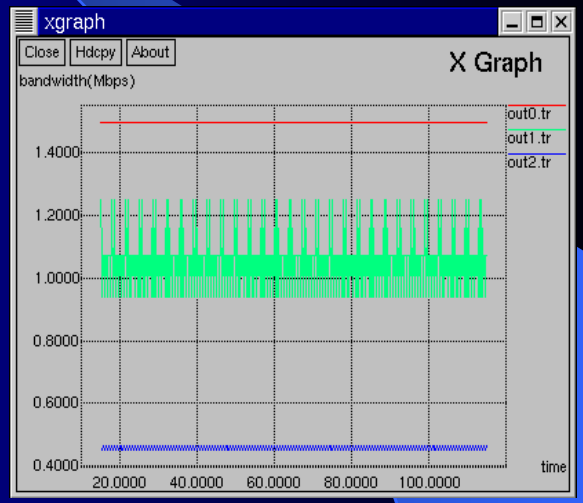
Reno-SACK



Reno-Vegas



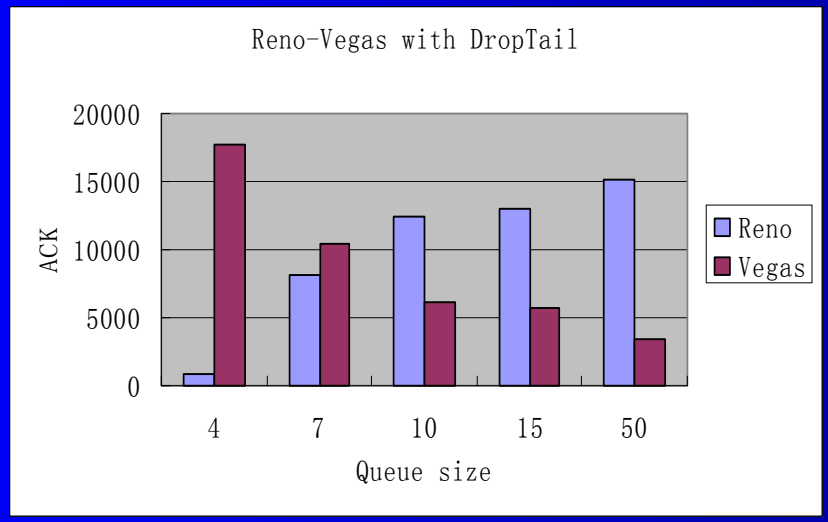
SACK-Vegas

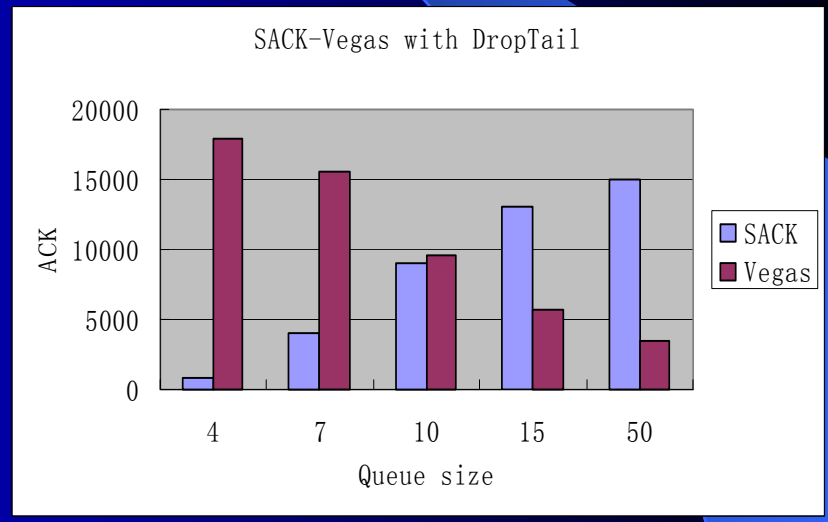


Retransmit: R-V 145/13170 vs. 0/5723
S-V 142/13208 vs. 0/5682

Queue Algorithm Effect

Fairness over Buffer Size Changes





Results Discussion

- Bandwidth Utilization of Vegas is High
- Reno and SACK Bias on Long Delay Link
- Vegas does not receive a fair share of bandwidth in the presence of Reno and SACK unless the buffer size are extremely small
- DropTail Buffer size will affect the fairness
- Other methods to cure the bias?

References

- [1] Kevin Fall, Sally Floyd, Simulation-based comparisons of Tahoe, Reno and SACK TCP, ACM SIGCOMM Computer Communication Review, v.26 n.3, p.5-21
- [2] S. Floyd, Congestion Control Principles, RFC2914, September 2000
- [3] L. Brakmo, S. O'Malley, and L. Peterson. TCP Vegas: New techniques for congestion detection and avoidance. In Proceedings of the SIGCOMM '94 Symposium (Aug. 1994) pages 24-35
- [4] S. Floyd, J. Mahdavi, M. Mathis, M. Podolsky, An Extension to the Selective Acknowledgement (SACK) Option for TCP, RFC2883, July 2000
- [5] V. Jacobson, Congestion avoidance and control, ACM SIGCOMM Computer Communication Review, v.18 n.4, p.314-329, August 1988
- [6] Jeonghoon Mo, Richard J. La, Venkat Anantharam, and Jean Walrand, Analysis and Comparison of TCP Reno and Vegas

Thank you