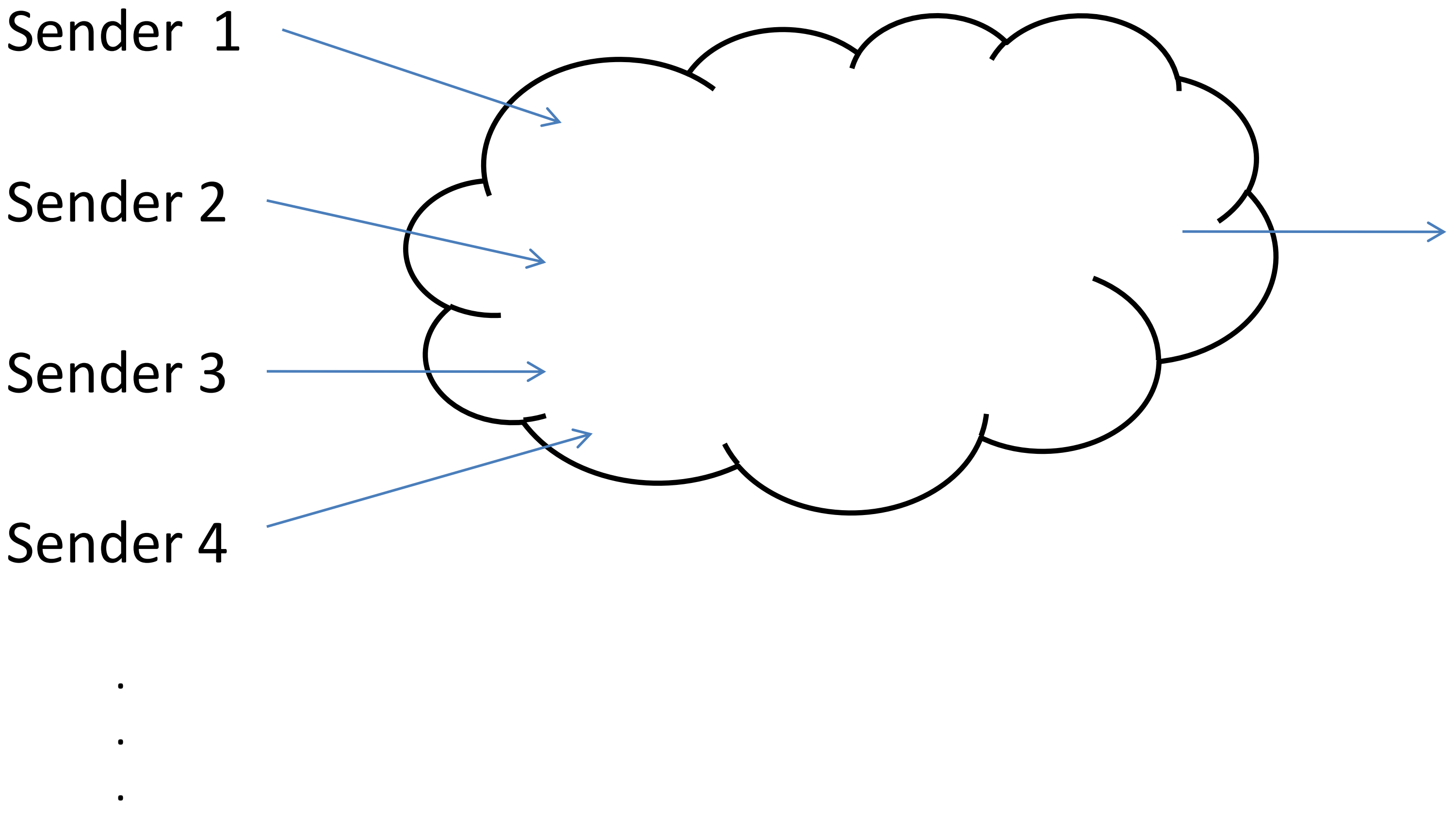
Recent Research in Congestion Control

The problem of high bandwidth-delay product connections

By Guillaume Marceau
Presented for WPI CS577, Advanced Computer Networks
December 8, 2009

The problem of congestion control

at which speed should Sender X inject data into the network?

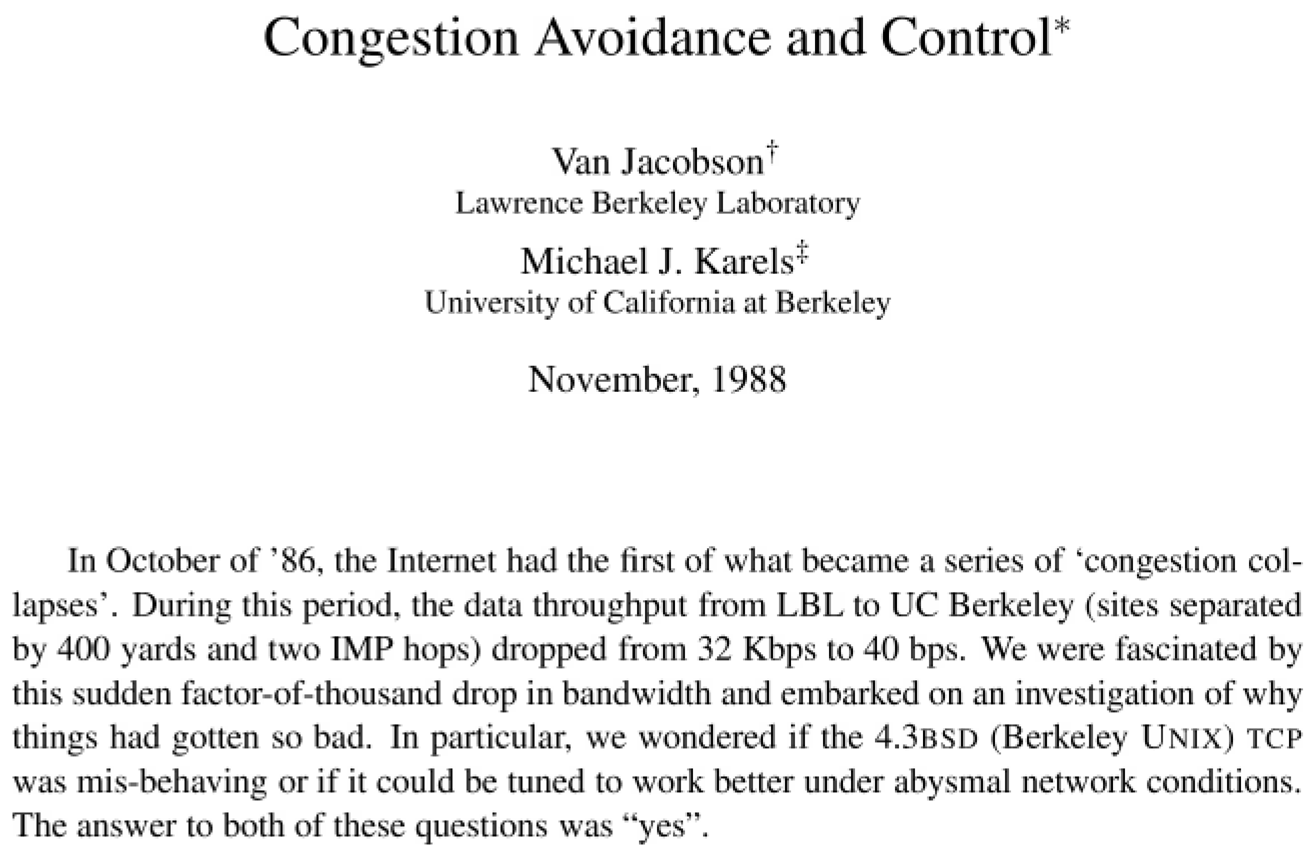


The problem of congestion control

at which speed should Sender X inject data into the network?

- No central knowledge of the capacity of the equipment installed
- No central knowledge of the topology
- The answer depends on the total demands
- No central coordination point between senders

- Everything varies: demands, topology, capacity



Congestion Avoidance and Control*

Van Jacobson†
Lawrence Berkeley Laboratory

Michael J. Karels‡
University of California at Berkeley

November, 1988

In October of '86, the Internet had the first of what became a series of 'congestion collapses'. During this period, the data throughput from LBL to UC Berkeley (sites separated by 400 yards and two IMP hops) dropped from 32 Kbps to 40 bps. We were fascinated by this sudden factor-of-thousand drop in bandwidth and embarked on an investigation of why things had gotten so bad. In particular, we wondered if the 4.3BSD (Berkeley UNIX) TCP was mis-behaving or if it could be tuned to work better under abysmal network conditions. The answer to both of these questions was "yes".

TCP Reno (1988)

- On each acknowledgement received in a round trip time in which congestion has not been detected

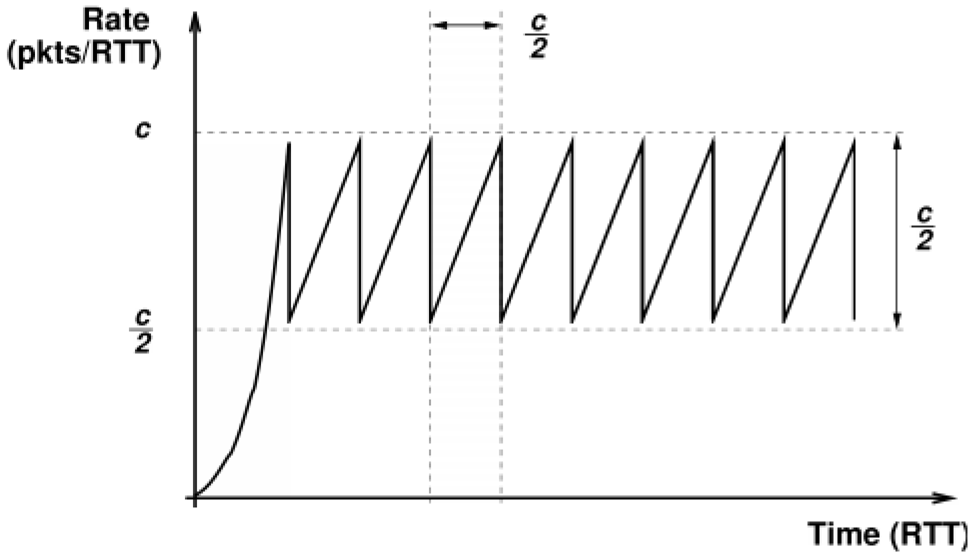
cwnd <- cwnd + 1/cwnd

- On the first detection of congestion in a given round trip time

cwnd <- cwnd – 0.5 * cwnd

- This is called *Additive Increase, Multiplicative Decrease* (AIMD)

TCP Reno (1988)



[from Scalable TCP, by Tom Kelly]

The problem of large bandwidth-delay product

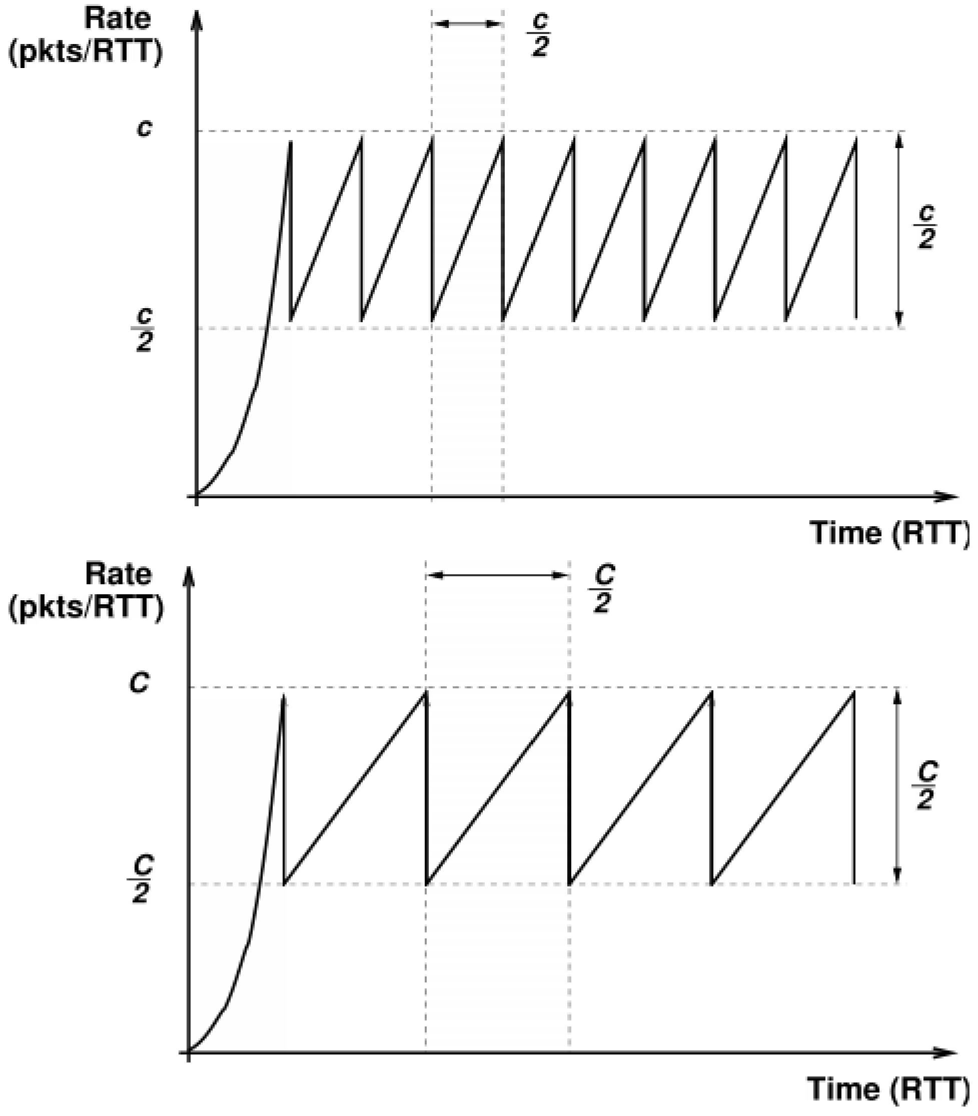


Figure 1: Traditional TCP scaling properties.

The problem of large bandwidth-delay product

an example

- 1 Gbps bandwidth, 200 ms round trip time
- Bandwidth-delay product of 17000 packets
- After a congestion event, cwnd = 8500
- It takes 8500 round trip times to re-acquire 1 Gbps

- 8500 * 0.2 second = 1700 seconds. That’s 28 minutes!

Requires extremely low loss rate

The essence of the problem

- Congestion control is a search problem
- You get some feedback once every RTT
- It’s only 1 bit of feedback
- And it’s noisy

Sources of noise

- Wireless (drops do not imply congestion)
- Background traffic
- Reverse traffic
- Oscillations

Stay at 1 bit - be more aggressive

Find new sources of information

Query the capacity explicitly

Be more aggressive – AIMD becomes MIMD

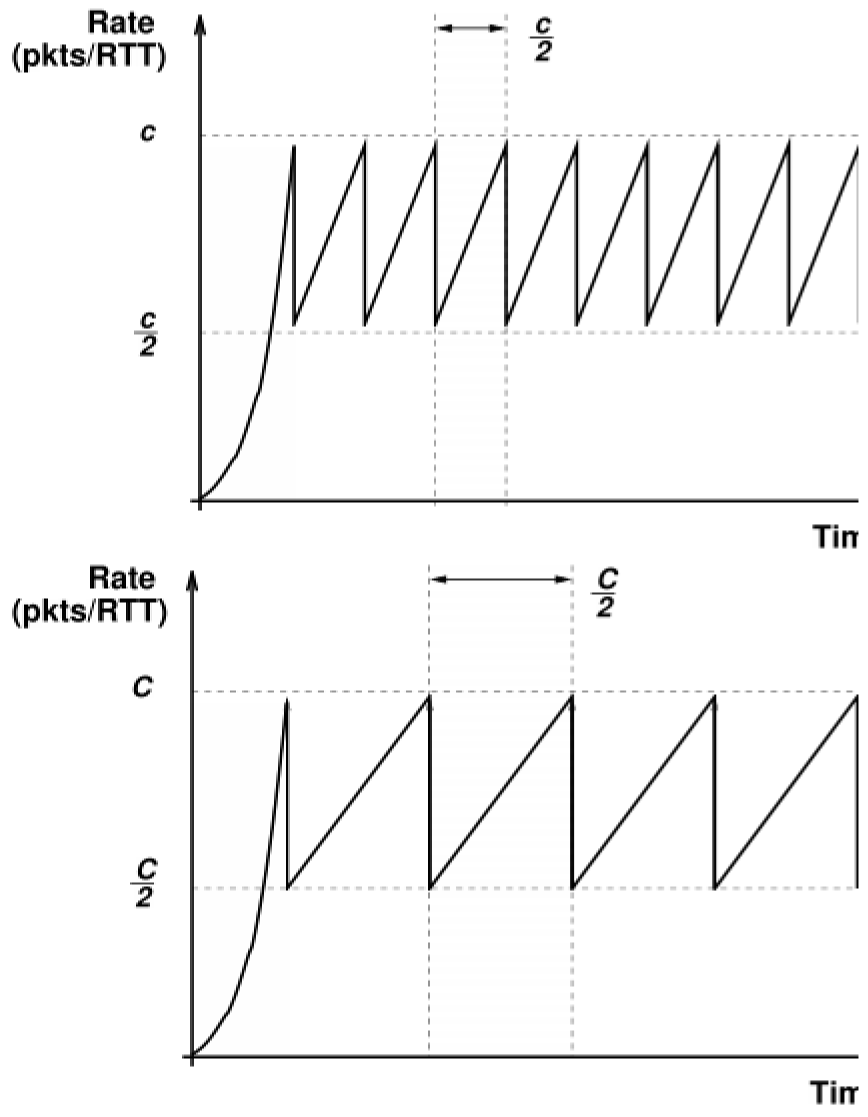


Figure 1: Traditional TCP scaling properties.

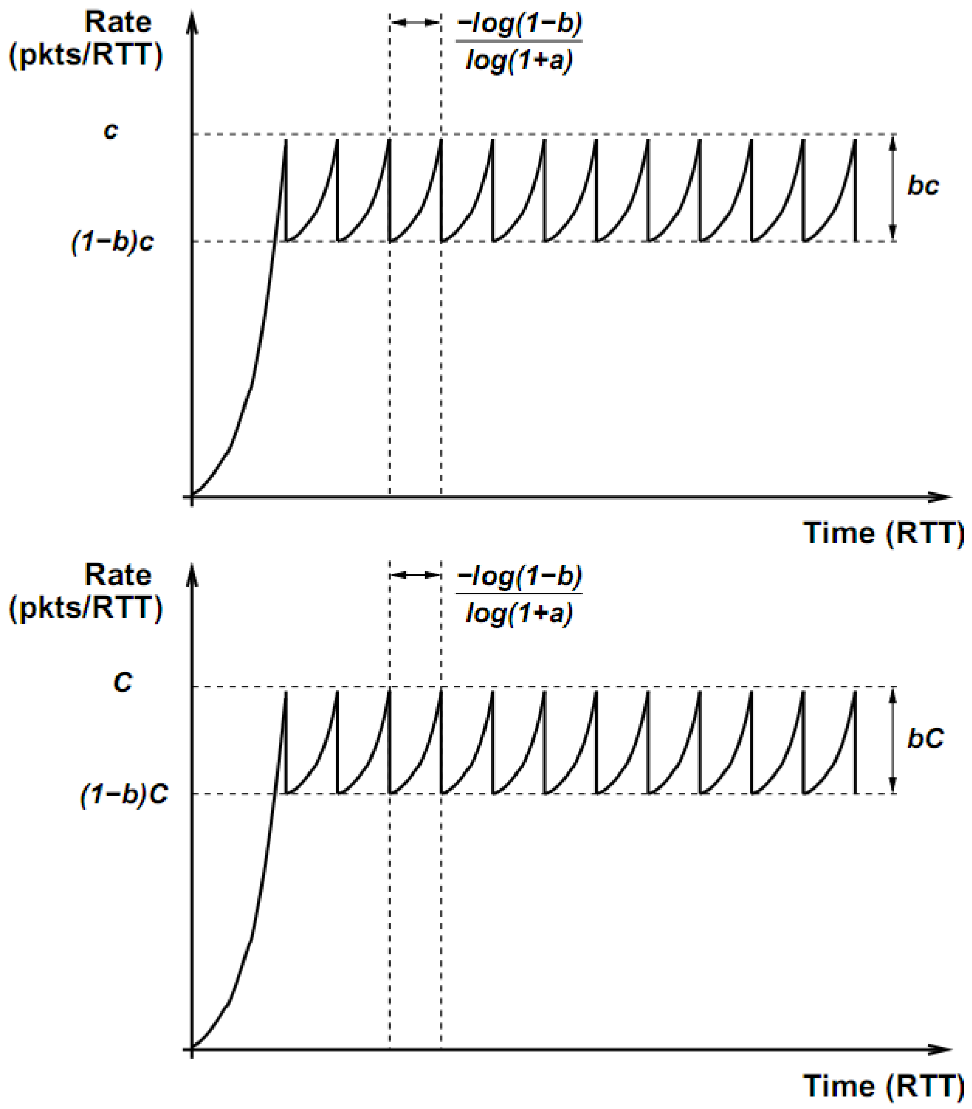


Figure 2: Scalable TCP scaling properties.

Be more aggressive

cwnd <- cwnd + A

cwnd <- cwnd – B * cwnd

Be more aggressive

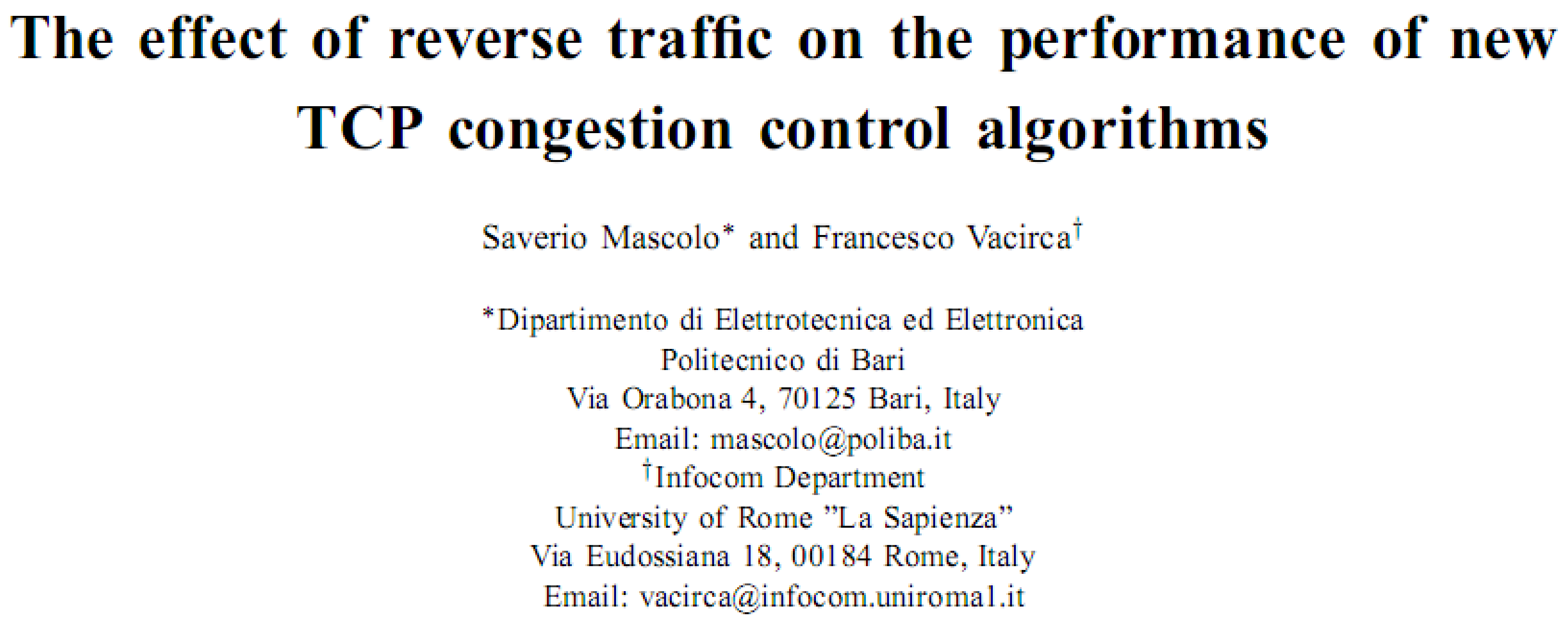
TCP Westwood,	‘01
TCP Westwood+	‘02
Scalable TCP	‘03
High-speed TCP	‘03
Hamilton TCP	‘04
BIC	‘04
CUBIC	‘05

More Information

delay	
Vegas	‘94
Fast	‘04
drop + delay	
Compound	‘06
Fusion	’07
other sources	
Peach	
Peach+	
RAPID	‘09

Query Explicitly

XCP	‘02
XCP-b	‘06
VCP	‘08
BMCC	‘09



The effect of reverse traffic on the performance of new TCP congestion control algorithms

Saverio Mascolo* and Francesco Vacirca†

*Dipartimento di Elettrotecnica ed Elettronica
Politecnico di Bari
Via Orabona 4, 70125 Bari, Italy
Email: mascolo@poliba.it
†Infocom Department
University of Rome "La Sapienza"
Via Eudossiana 18, 00184 Rome, Italy
Email: vacirca@infocom.uniroma1.it

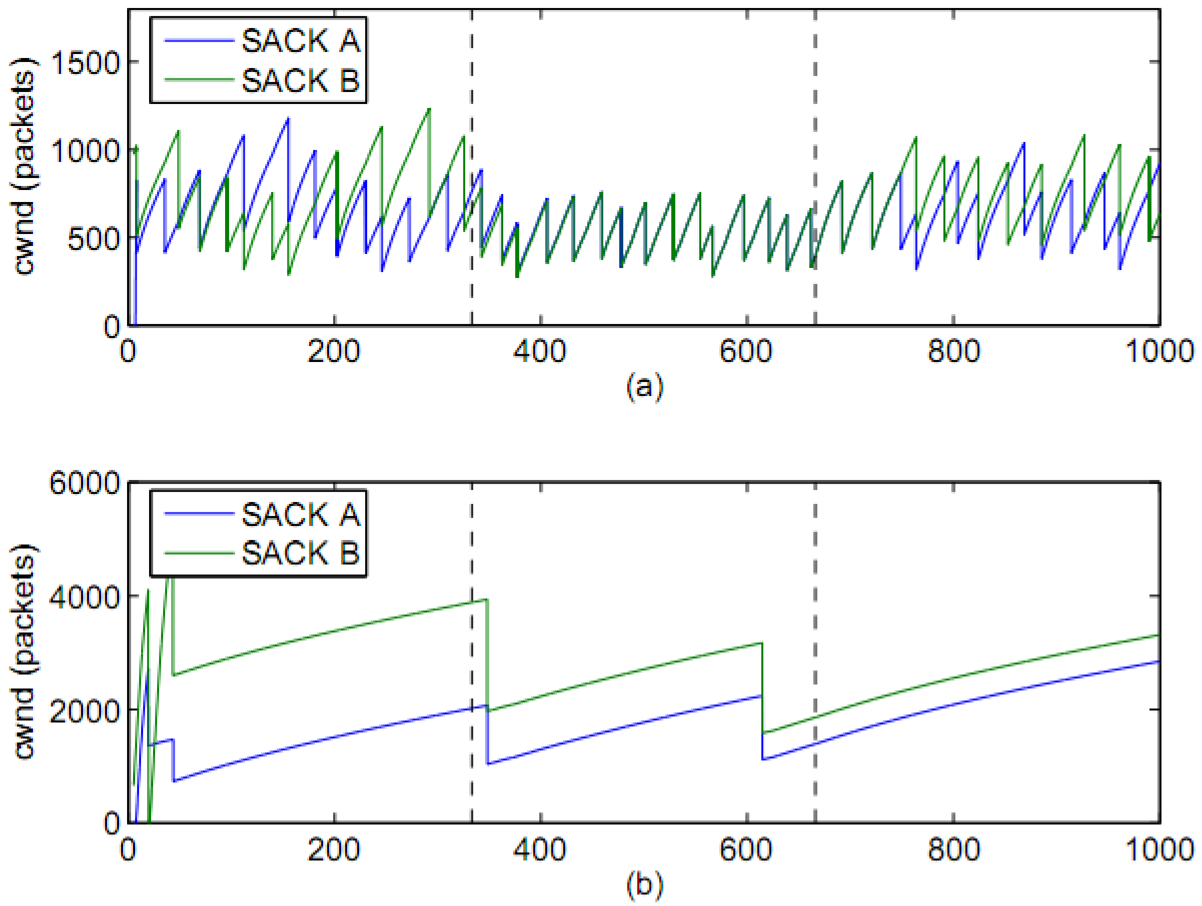


Fig. 6. SACK TCP: Congestion window evolution with Web traffi c enabled: (a) RTT=40ms and (b) RTT=160ms.

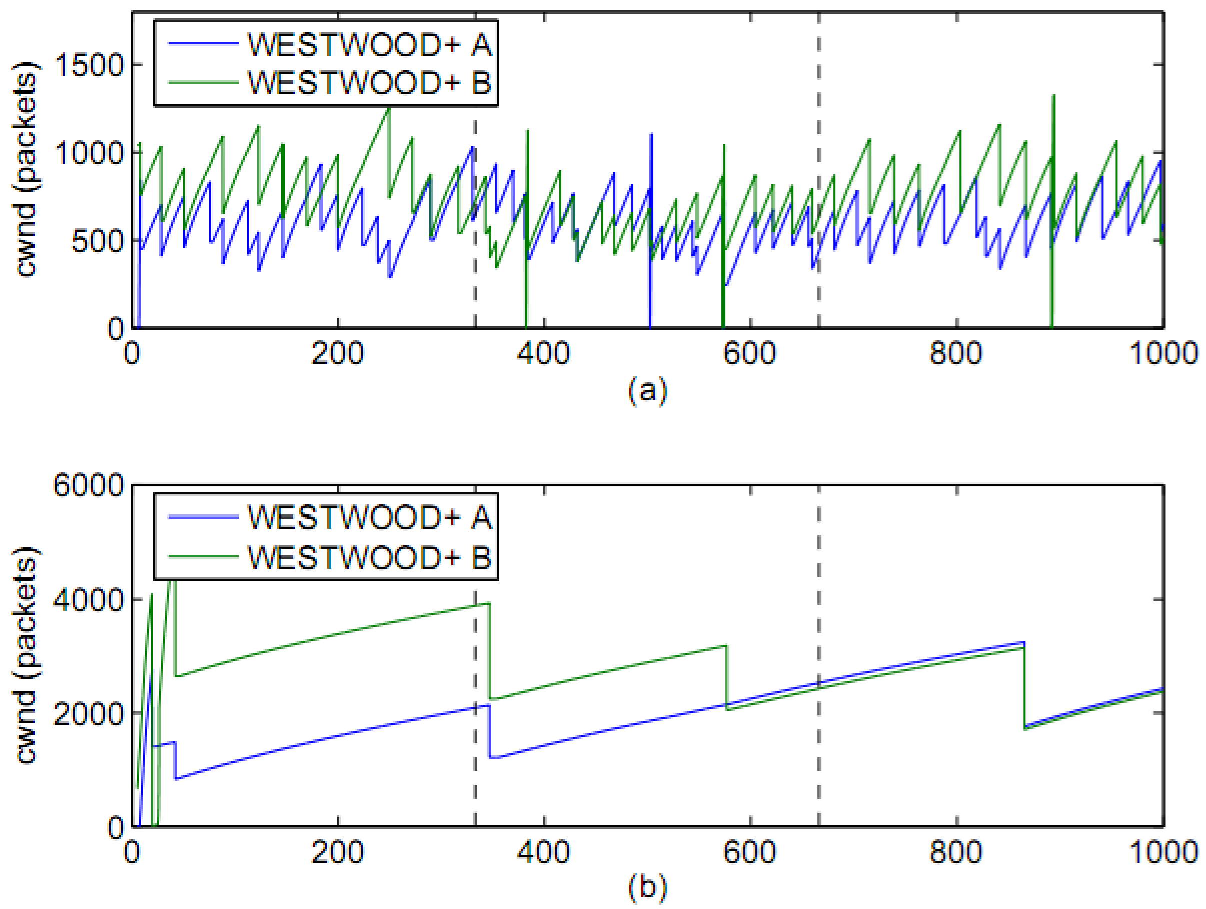


Fig. 11. Westwood+ TCP: Congestion window evolution with Web traffi c enabled: (a) RTT=40ms and (b) RTT=160ms.

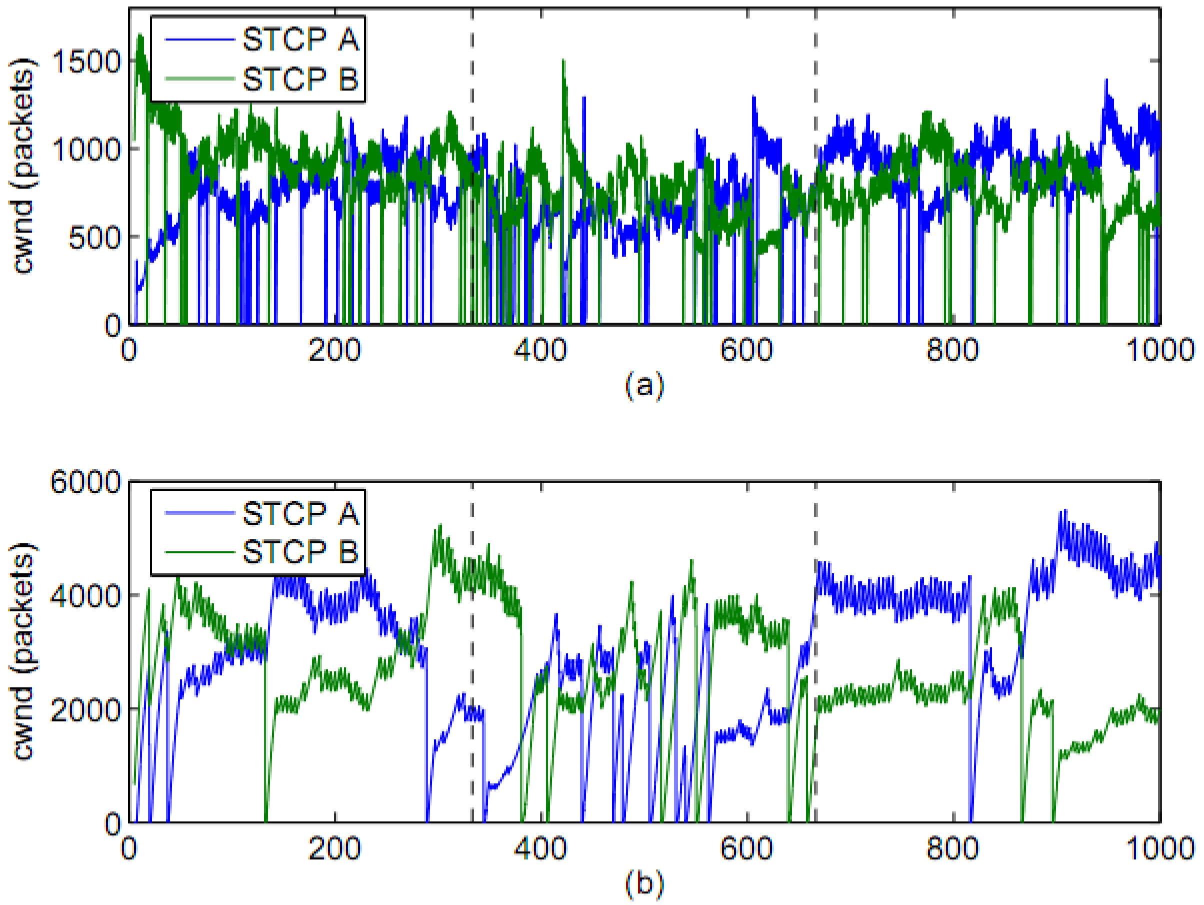


Fig. 10. STCP: Congestion window evolution with Web traffi c enabled: (a) RTT=40ms and (b) RTT=160ms.

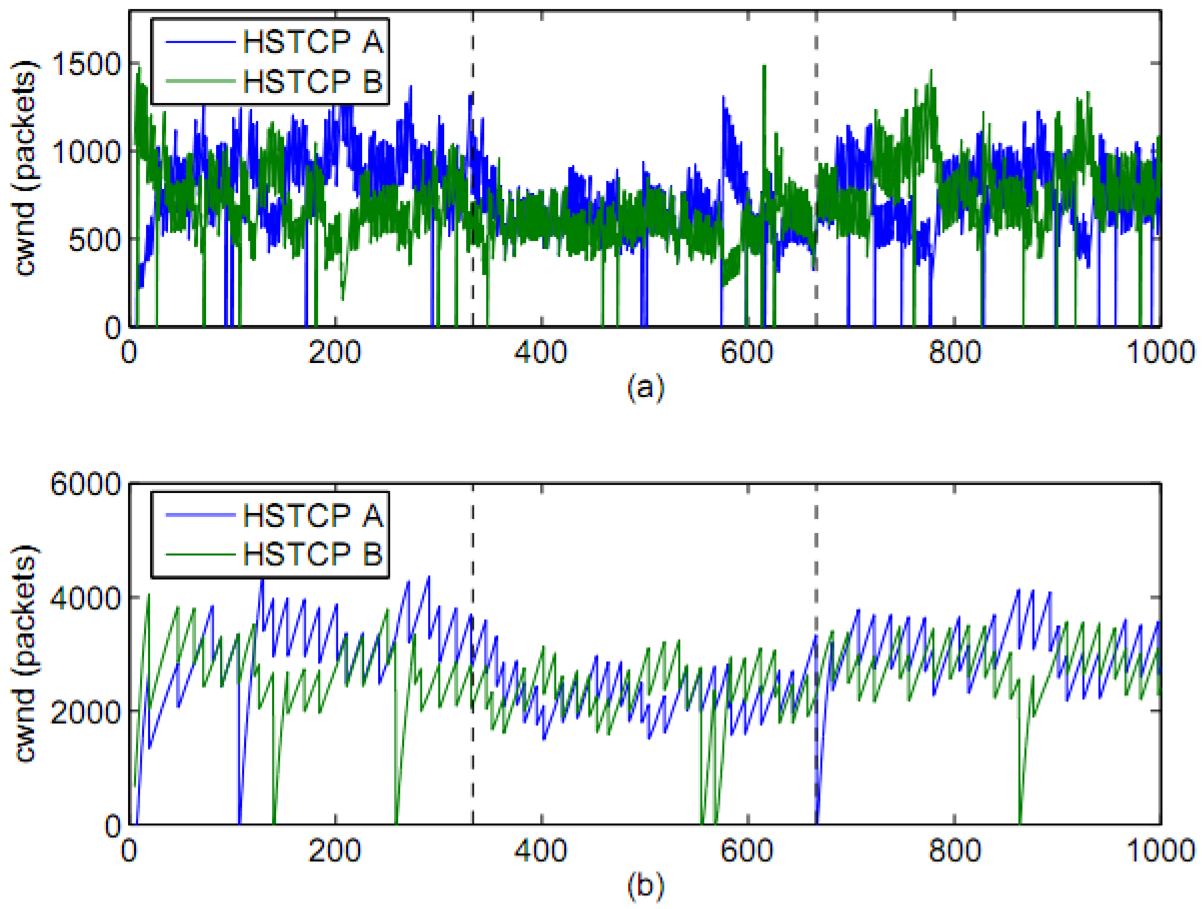


Fig. 7. HSTCP: Congestion window evolution with Web traffi c enabled: (a) RTT=40ms and (b) RTT=160ms.

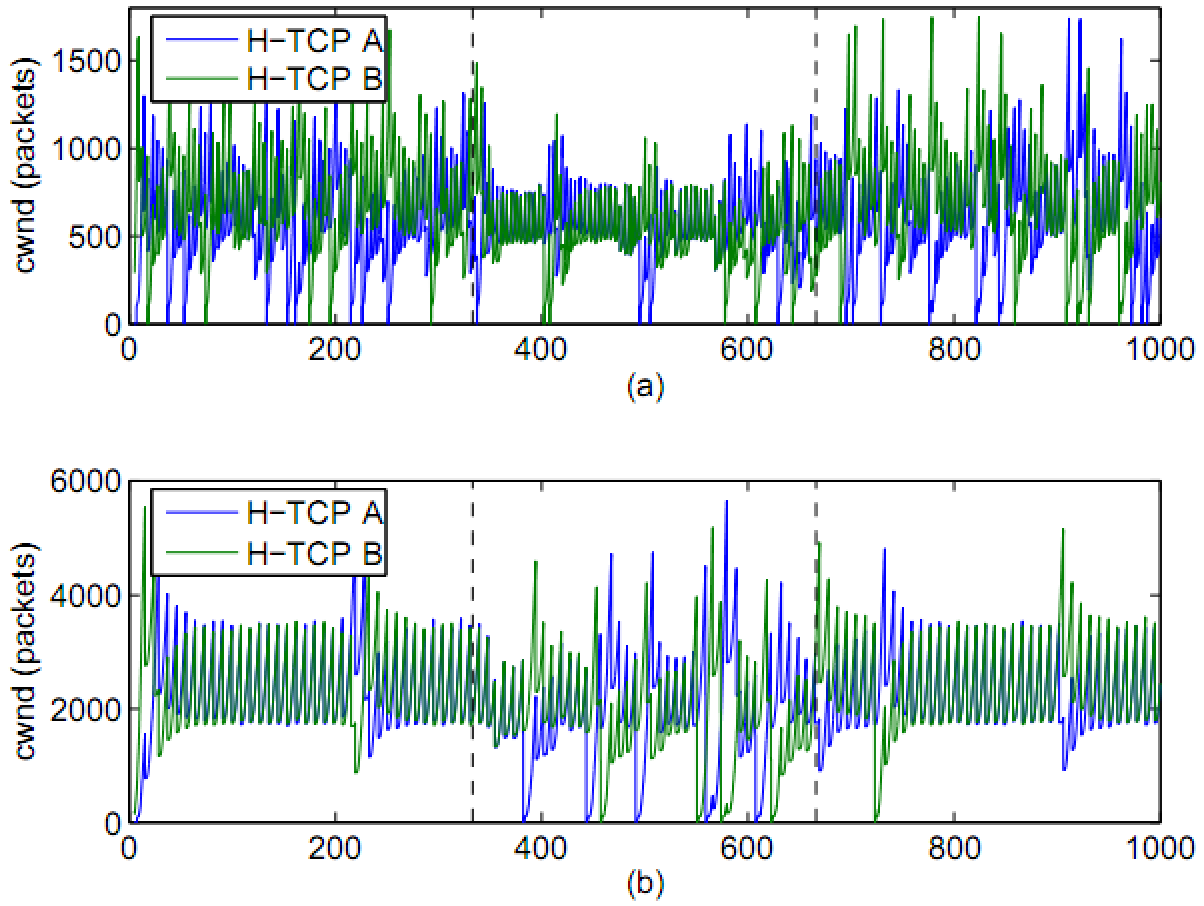


Fig. 8. H-TCP: Congestion window evolution with Web traffi c enabled: (a) RTT=40ms and (b) RTT=160ms.

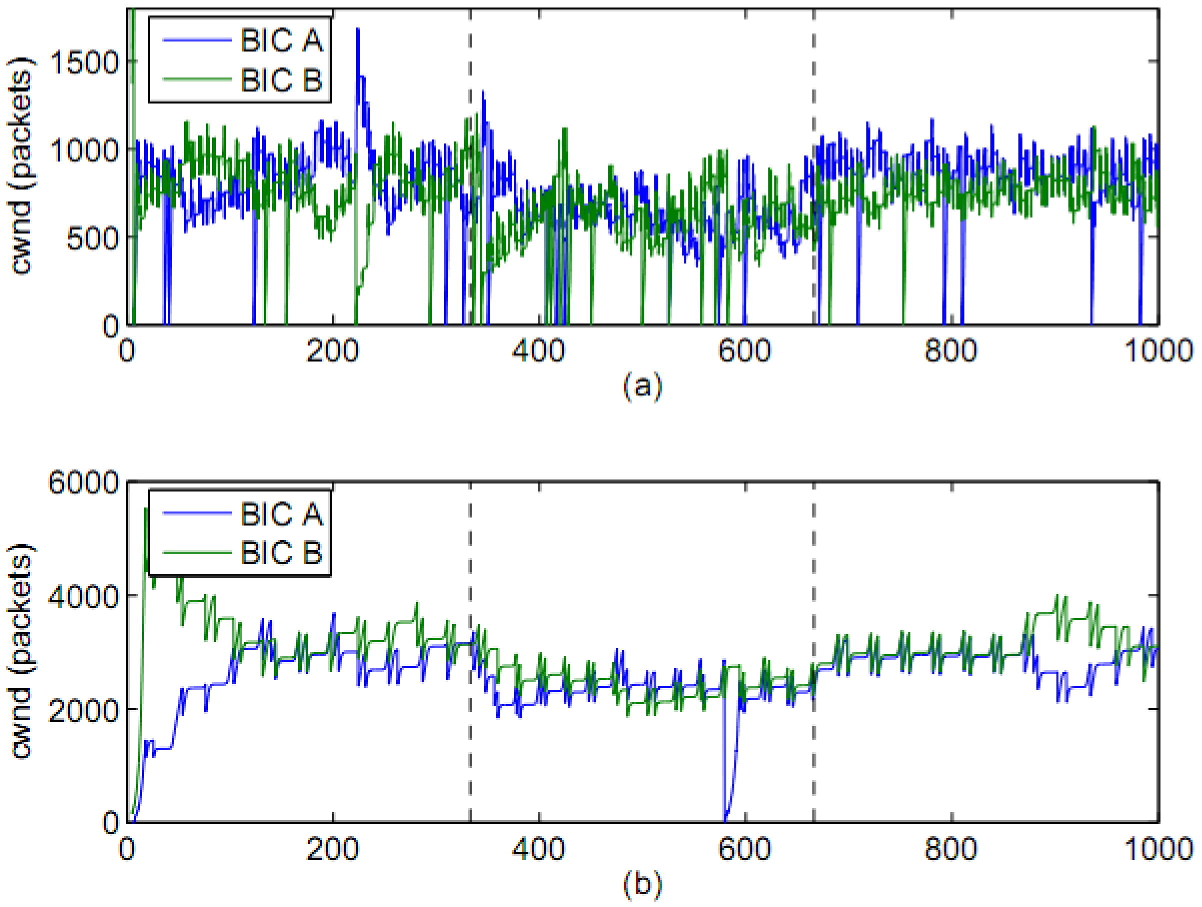
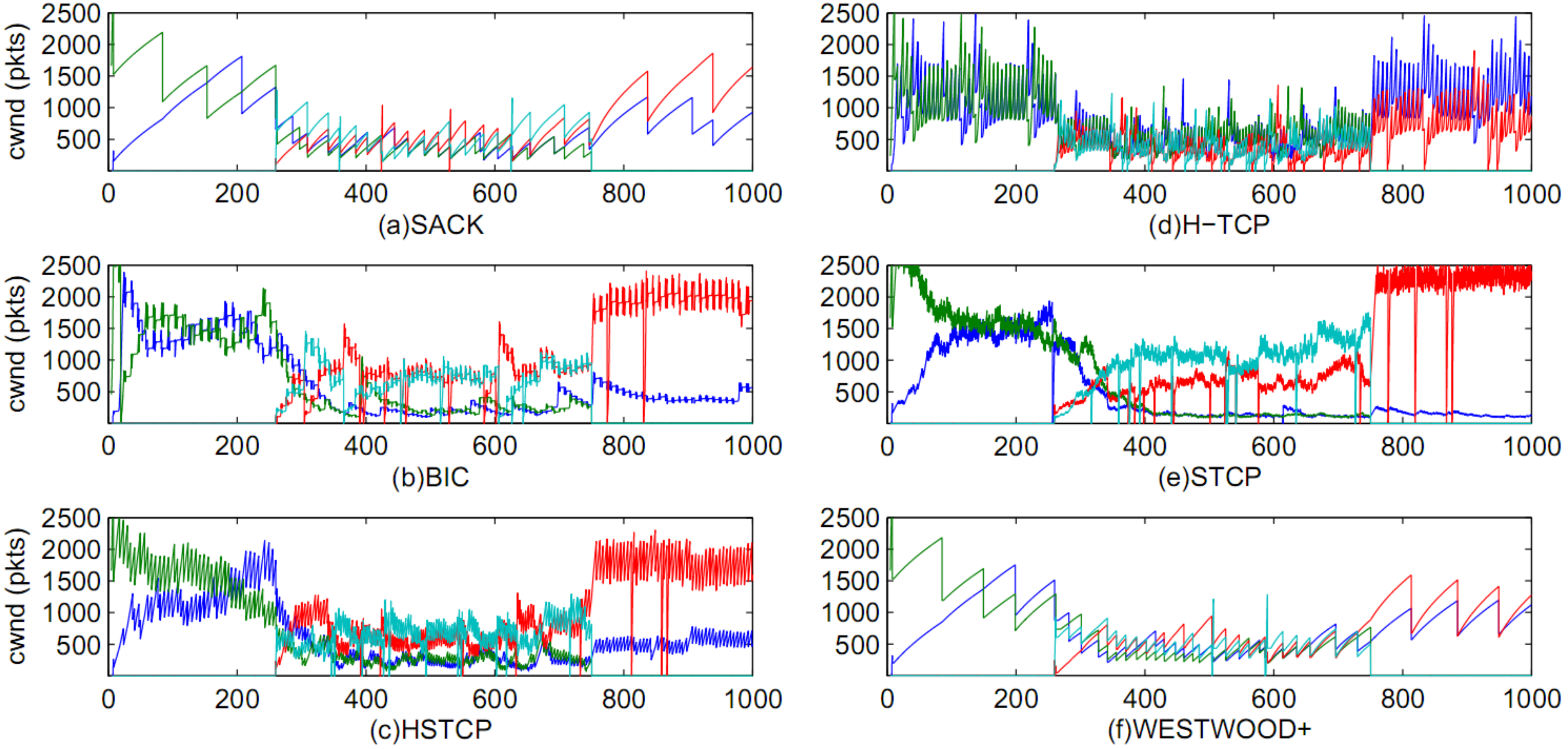


Fig. 9. BIC: Congestion window evolution with Web traffi c enabled: (a) RTT=40ms and (b) RTT=160ms.



Dumbbell topology, reserve traffic, background Web traffic, **different RTT** [from Mascolo nd Vacirca, 2007]

Be more aggressive -- Conclusions

- Quickly leads to instabilities
- Clobbers less aggressive streams (Reno)
- RTT unfairness remains

More information – Delay-based

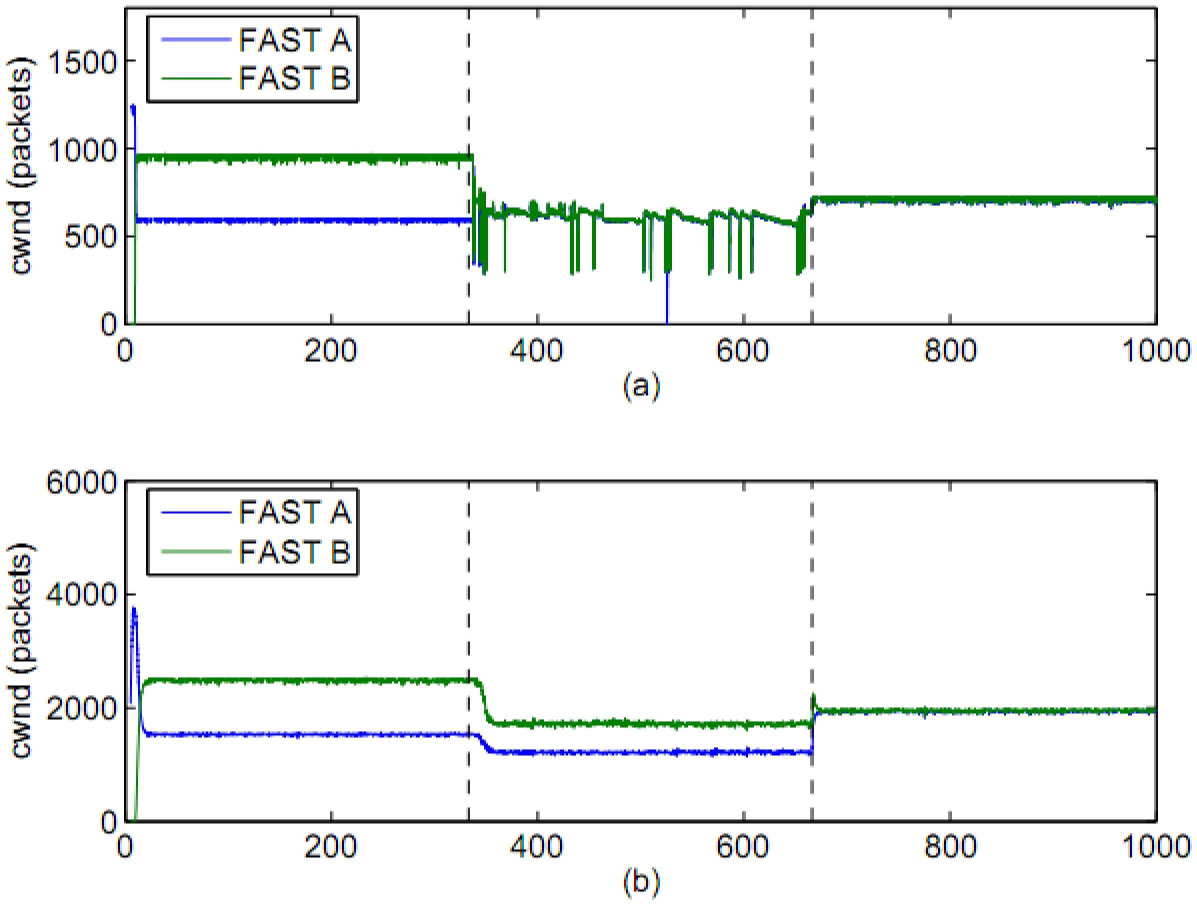


Fig. 12. FAST TCP: Congestion window evolution with Web traffi c enabled: (a) RTT=40ms and (b) RTT=160ms.

More information – FAST TCP

by Chen Jin, David X. Wei, Steven H. Low, at Caltech

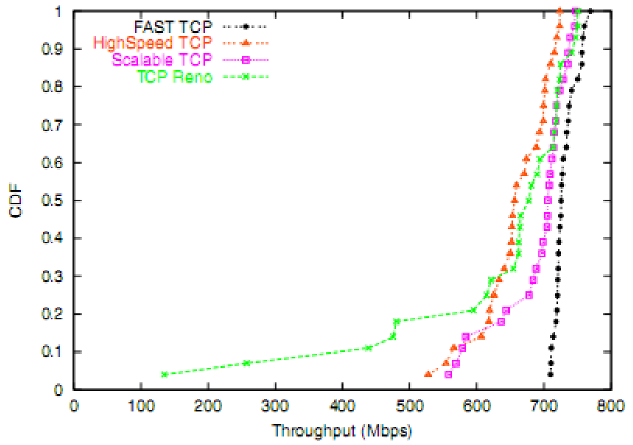


Fig. 12. Overall evaluation: throughput.

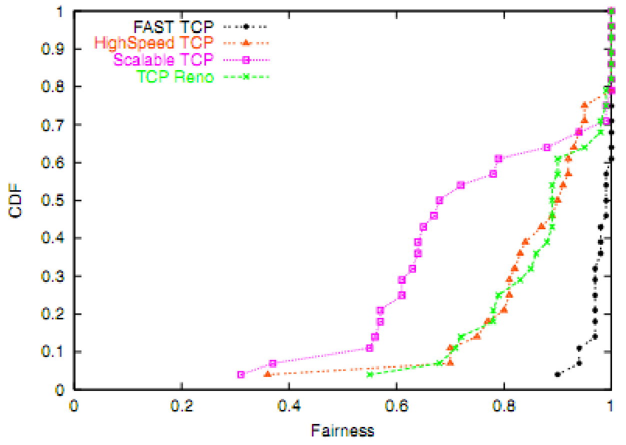


Fig. 13. Overall evaluation: fairness.

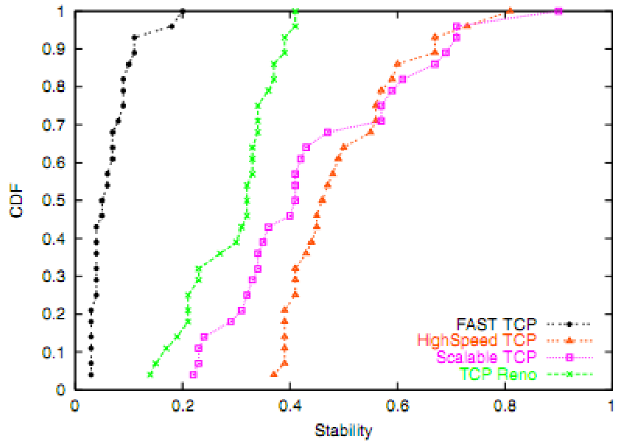


Fig. 14. Overall evaluation: stability.

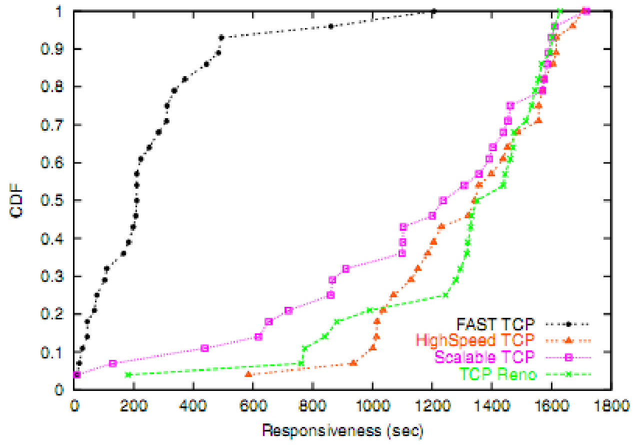


Fig. 15. Overall evaluation: responsiveness index R_1.

More information – Conclusion

- Delays alone is too mild -- they gets clobbered by Reno
- TCP Fusion and TCP Compound mix loss & delay

The Internet Today

- Linux
  - Has used TCP CUBIC since 2.6.19 (Nov 2006)
- Windows
  - Has used TCP Compound since Vista

Be more aggressive

TCP Westwood,	‘01
TCP Westwood+	‘02
Scalable TCP	‘03
High-speed TCP	‘03
Hamilton TCP	‘04
BIC	‘04
CUBIC	‘05

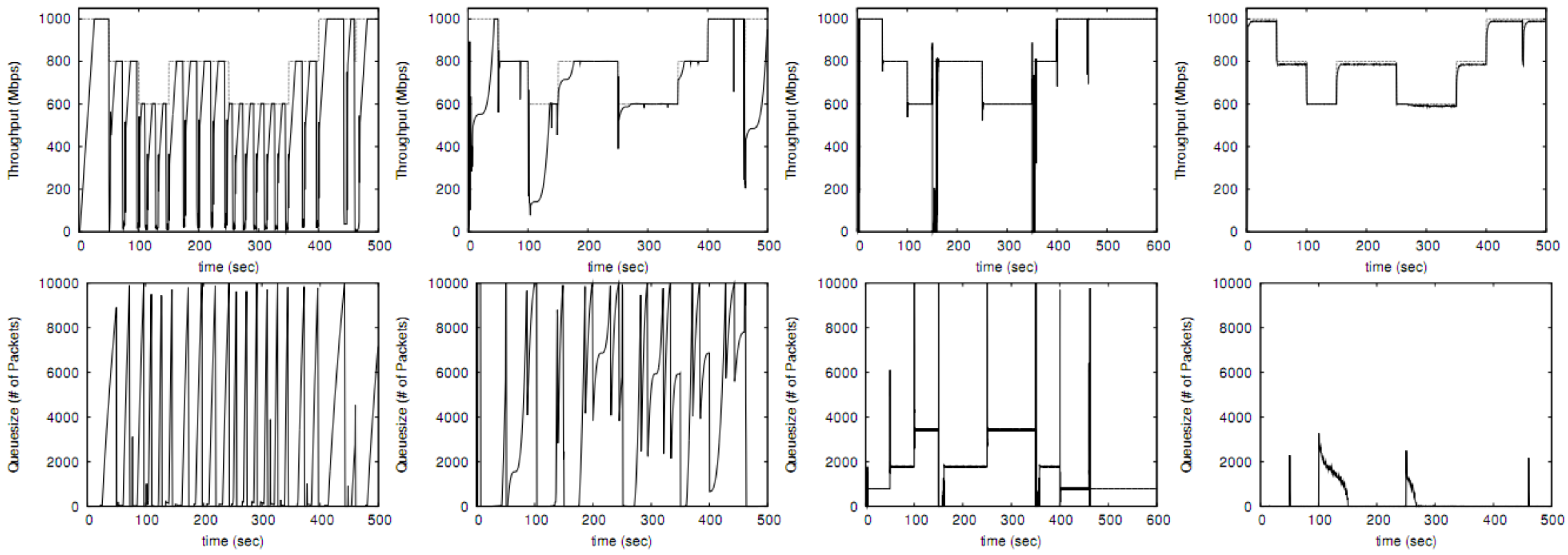
More Information

delay	
Vegas	‘94
Fast	‘04
drop + delay	
Compound	‘06
Fusion	’07
other sources	
Peach	‘01
Peach+	‘02
RAPID	‘09

Query Explicitly

XCP	‘02
XCP-b	‘06
VCP	‘08
BMCC	‘09

More information -- RAPID

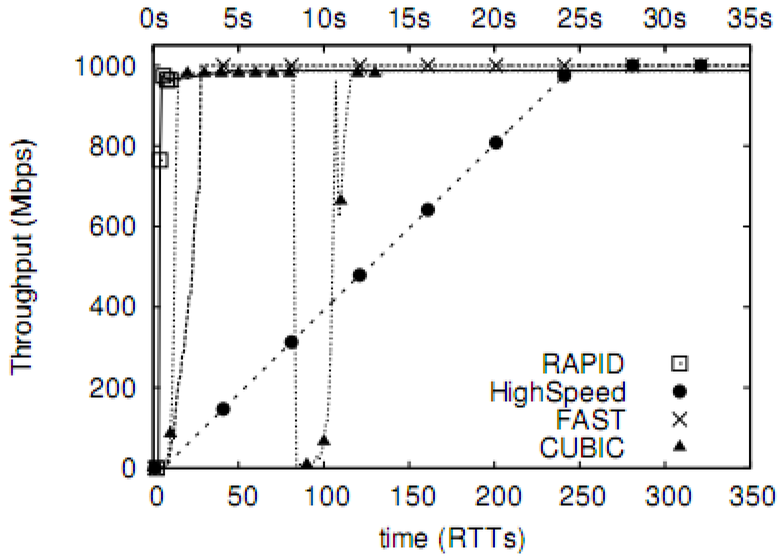


(a) HighSpeed (b) CUBIC (c) FAST (d) RAPID

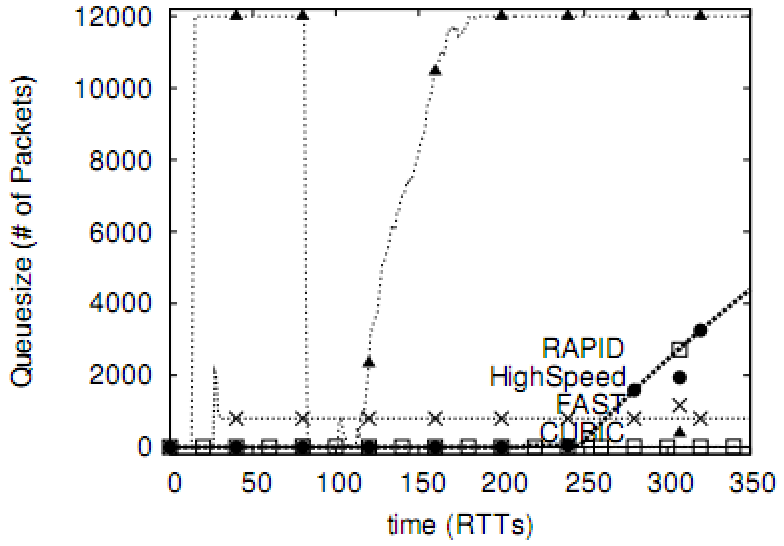
Fig. 4. Performance in a Dynamic Bandwidth 1 Gbps Network

RAPID: Shrinking the Congestion-control Timescale
Vishnu Konda and Jasleen Kaur
University of North Carolina at Chapel Hill

More information -- RAPID



(a) Throughput vs. Time



(b) Bottleneck Queue vs. Time

Fig. 3. Performance of Slow Start on a 1 Gbps Network

Conclusions

- A lot of effort spent tuning drop-based congestion control (with no definite result)
- Obtaining additional sources of information helps a lot
- Explicit congestion control is now feasible