High-Speed TCP: Recent Developments, Issues and Challenges

Raj Jain
Washington University In Saint Louis
Saint Louis, MO 63131
Jain@wustl.edu

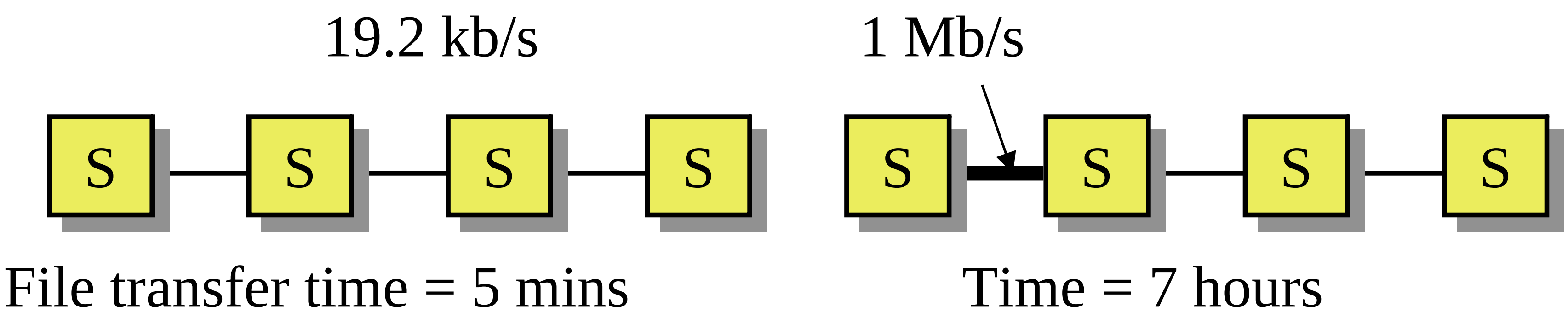
Microsoft High-Speed TCP Workshop
Redmond, WA, February 4-5, 2007

These slides are also available on-line at
http://www.cse.wustl.edu/~jain/talks/mstcp.htm

1. Our Congestion Research
2. Then vs Now (1980’s vs 2000’s)
3. High-Speed TCPs
4. Top 10 Requirements for a Good Scheme
5. Two New Problems for Congestion experts

Our Congestion Research

- 1979-1980: High-Speed Network = 10Mbps Ethernet



- Implicit Congestion Indication: Drop ⇒ Congestion. Drop window to 1.
- How to adjust windows: AIMD
- Explicit Congestion Indication: DECBit
- April 1987: ARPA INENG (IETF)- Bit in the packet header, Increase/Decrease
- August 1988: Slow start paper by V. Jacobson

IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL. SAC-4, NO. 7, OCTOBER 1986

A Timeout-Based Congestion Control Scheme for Window Flow-Controlled Networks

RAJ JAIN, SENIOR MEMBER, IEEE

Abstract—During overload, most networks drop packets due to buffer unavailability. The resulting timeouts at the source provide an implicit mechanism to convey congestion signals from the network to the source. On a timeout, a source should not only retransmit the lost packet, but it should also reduce its load on the network. Based on this realization, we have developed a simple congestion control scheme using the acknowledgment timeouts as indications of packet loss and congestion. This scheme does not require any new message formats, therefore, it can be used in any network with window flow control, e.g., ARPAnet or ISO.

Increase Policy

4) *Increase:* *WS* can be increased by one after the number of packets acknowledged since the last change (increase or decrease) becomes greater than or equal to the current value of *WS*. This gives a parabolic rise to *WS* when plotted against packets acknowledged. Notice however, the rise is approximately linear in time because with *n* packets outstanding, it takes one round-trip delay to get an acknowledgment for the *n* packets. Thus, *WS* increases by one every round-trip delay interval.

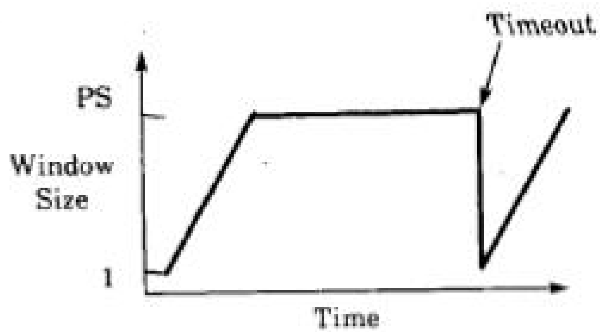


Fig. 1. Dynamic window adjustment using CUTE scheme.

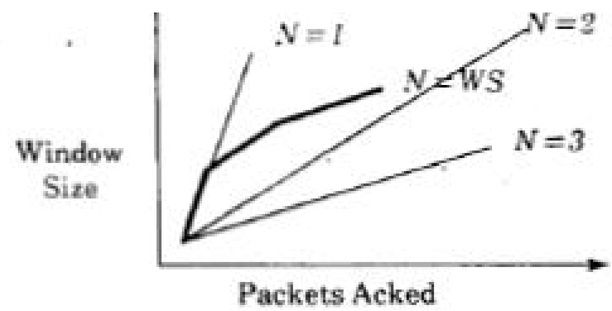
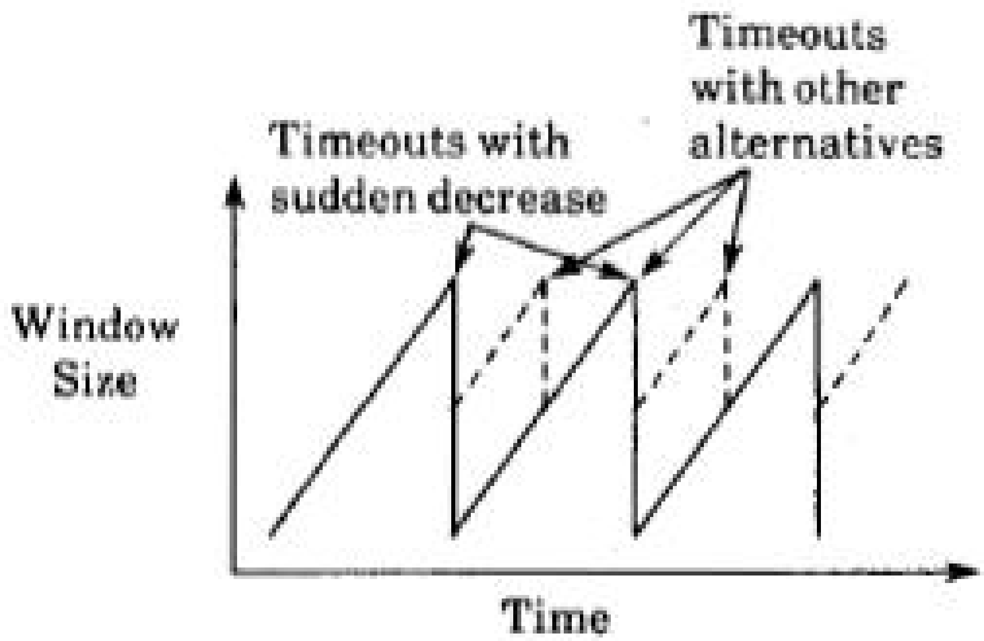


Fig. 5. Increase policies considered.

Decrease Policy

5) *Decrease:* On a timeout, the source should reset *WS* to the minimum allowed value.

$$WS \leftarrow WS_{min}$$



Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks

Dah-Ming CHIU and Raj JAIN
Digital Equipment Corporation, 550 King Street (LKG1-2/A19), Littleton, MA 01460-1289, U.S.A.

Proposition 3. *For both feasibility and optimal convergence to fairness, the increase policy should be additive and the decrease policy should be multiplicative.*

North-Holland
Computer Networks and ISDN Systems 17 (1989) 1–14

AIMD

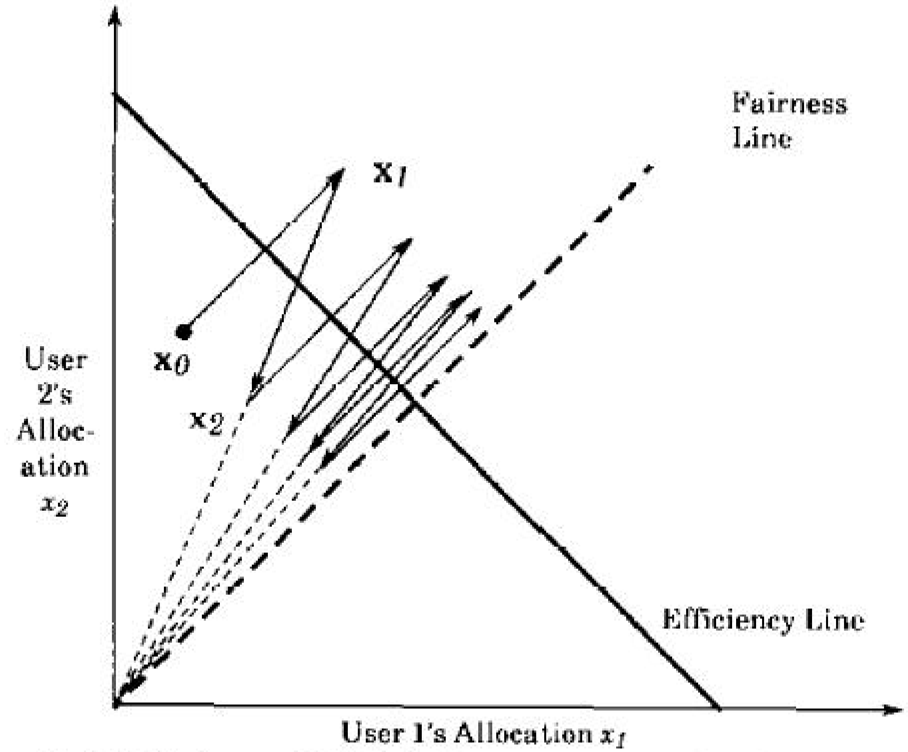


Fig. 5. Additive Increase/Multiplicative Decrease converges to the optimal point.

Where were We Then?

- 10 Mbps $\times$ 2 km $= 10 \times 10^{6} \times 2 \times 10^{3} \times 5 \times 10^{-6} \times 2$ $= 200{,}000$ bits $= 25{,}000$ bytes $= 17$ 1500B-packets.
- Store and forward delays >> propagation delays $\Rightarrow$ Usual window = 8
- How you go from initial window to 8 has some minor effect
- How you come down had a major effect

Where are We Now?

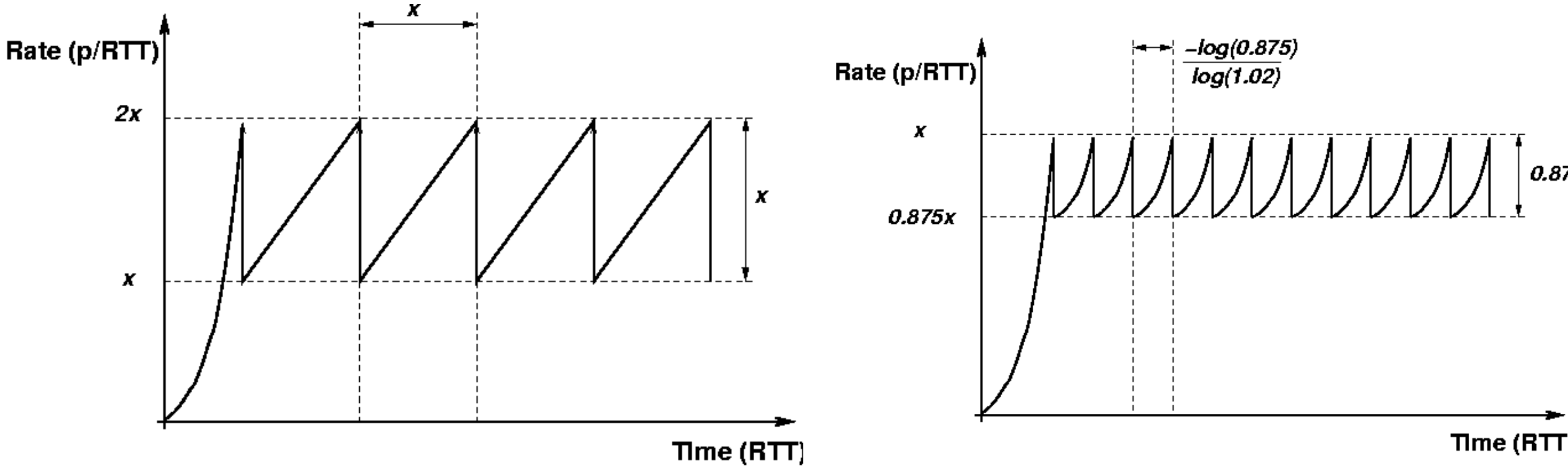
- 1G is on the laptop/desktop
- 10G is common in data center
- 100G×40km Ethernet is being standardized in IEEE 802.3
- n×10G is used in metro networks via Link Aggregation
- 10G × 4834 miles coast-to-coast
  $= 10^{10} \times 4834 \times 1.6 \times 5 \times 10^{-6} \times 2$ bits
  = 773.44 Mb = 96 MB = 198000 512B-segments
- Which ever way you count from 1 to 198,000 is going to be slow...

LFNs (Elephens)

- RFC1072 (October 1988):
  - LFNs > 10^5 bits = 12 kB = 8 1500B-packets
- TCP needs receive window = BDP to keep the pipe full
- Ideal Sender window size = 2BDP to recover from errors
- You need to send at least 3BDP bytes to get to full speed
- Bandwidth = Receive window/RTT
- Default TCP buffer size = 64 kB
- 64 kB window, 200ms RTT
  $\Rightarrow$ Max rate = 64kB/200ms = 2.5 Mbps

High-Speed TCPs

- **Core Problem**: TCP Reno increases its rate too slowly and decreases it too fast.
- **Solution**: Rise faster and come down slower than Reno



High-Speed TCPs

- **HS-TCP**, Sally Floyd, http://www.icir.org/floyd/hstcp.html
- **Scalable TCP**, Tom Kelly, http://www-lce.eng.cam.ac.uk/~ctk21/scalable/
- **Fast TCP**, Steven Low, http://netlab.caltech.edu/FAST/
- **BIC/CUBIC**, Injong Rhee, http://www.csc.ncsu.edu/faculty/rhee/export/bitcp/
- **Layered TCP** (LTCP), http://students.cs.tamu.edu/sumitha/research.html
- **Hamilton TCP** (HTCP), http://www.hamilton.ie/net/htcp/
- **TCP Westwood**, Mario Gerla, http://www.cs.ucla.edu/NRL/hpi/tcpw/
- …

Most of these require only send-side modifications to TCP

Congestion in Datacenter Networks

- Bounded delay-bandwidth product
  - High-speed: 10 Gbps (now) 100 Gbps (future)
  - Short round-trip delays
  - 1 Mb to 5 Mb delay-bandwidth product
- Storage Traffic ⇒ short access times ⇒ Low delay
- Packet loss ⇒ Long timeouts ⇒ Not desirable
- IEEE 802.1au Congestion Notification

Top 10 Requirements for a Good Scheme

1. Fast convergence to stability in rates
2. Fast convergence to fairness. Proportional or Max-min

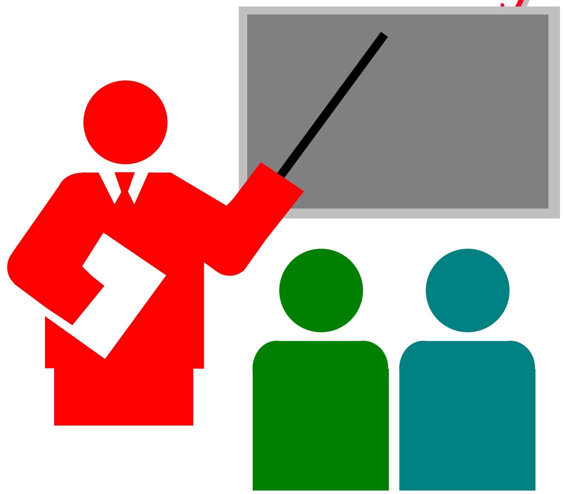
   $$\text{Fairness Index} = \frac{\Sigma_i^n x_i^2}{n(\Sigma_i^n x_i)^2} \qquad x_i = \frac{\text{Actual Allocation}}{\text{Fair Allocation}}$$

3. Good for bursty traffic ⇒ Fast convergence
4. Efficient operation: minimize unused capacity. Minimize chances of router Q=0 when sources have traffic to send
5. Extremely low (or zero) loss
6. Predictable performance: No local minima
7. Easy to deploy ⇒ Small number of parameters
8. Easy to set parameters
9. Parameters applicable to a wide range of network configurations link speeds, traffic types, number of sources.
10. Applicable to a variety of router architectures and queueing/scheduling disciplines

Transport for Internet 3.0

- Internet 3.0 is the next generation of Internet
  Internet 1.0 = First 20 years = ARPAnet (1969-89)
  Internet 2.0 = 2^{nd} 20 years = 1989-2009
- NSF GENI/FIND project
- How would you design a transport layer today?
  - Window vs Rate
  - Layered vs Cross-Layer
  - AIMD vs Explicit
  - Pacing: Removing Burstiness
- Ref: Raj Jain, "Internet 3.0: Ten Problems with Current Internet Architecture and Solutions for the Next Generation," http://www.cse.wustl.edu/~jain/papers/gina.htm

Summary



1. Time to transition from implicit feedback, AIMD, and window
2. Handling elephants is easier. Mice are challenging.
   Most of the internet flows are bursty.
3. Speed of convergence to stability and fairness is important for bursty traffic
4. Time to think about traffic management in the next generation Internet.

References

- J. Jiang and R. Jain, “Forward Explicit Congestion Notification (FECN) for Datacenter Ethernet Networks,” IEEE 802.1au interim Meeting, Monterrey, CA, January 24-26, 2007, http://www.cse.wustl.edu/~jain/ieee/fecn701.htm
- Raj Jain, “Internet 3.0: Ten Problems with Current Internet Architecture and Solutions for the Next Generation,” in Proceedings of Military Communications Conference (MILCOM 2006), Washington, DC, October 23-25, 2006, http://www.cse.wustl.edu/~jain/papers/gina.htm
- R. Jain, "A Timeout Based Congestion Control Scheme for Window Flow-Controlled Networks," IEEE Journal of Selected Areas in Communications, Vol. SAC-4, No. 7, October 1986, pp. 1162-1167. Reprinted in C. Partridge, Ed., "Innovations in Internetworking," Artech House, Norwood, MA 1988, http://www.cse.wustl.edu/~jain/papers/control.htm

References 2

- D. Chiu and R. Jain : Analysis of the Increase and Decrease Algorithms for Congestion Avoidance. in Computer Networks, Computer Networks and ISDN Systems, vol. 17, pp. 1–14, 1989, http://www.cse.wustl.edu/~jain/papers/cong_av.htm
- K. Ramakrishnan and R. Jain, "A Binary Feedback Scheme for Congestion Avoidance in Computer Networks with Connectionless Network Layer," Proc. SIGCOMM'88, August 1988, pp. 303-313, http://www.cse.wustl.edu/~jain/papers/cr2.htm
- R. Jain, D. Chiu, and W. Hawe, "A Quantitative Measure Of Fairness And Discrimination For Resource Allocation In Shared Computer Systems," DEC Research Report TR-301, September 1984, http://www.cse.wustl.edu/~jain/papers/fairness.htm