A Survey of Results on TCP/IP over ATM

Raj Jain

Raj Jain is now at
Washington University in Saint Louis
Jain@cse.wustl.edu
http://www.cse.wustl.edu/~jain/

Our Team

- **Current**:
  - Shivkumar Kalyanaraman
  - Rohit Goyal
  - Sonia Fahmy
  - Jianping Jian
- **Past**:
  - Fang Lu
  - Ram Viswanathan
  - Manu Vasandani
  - Arun Krishnamoorthy

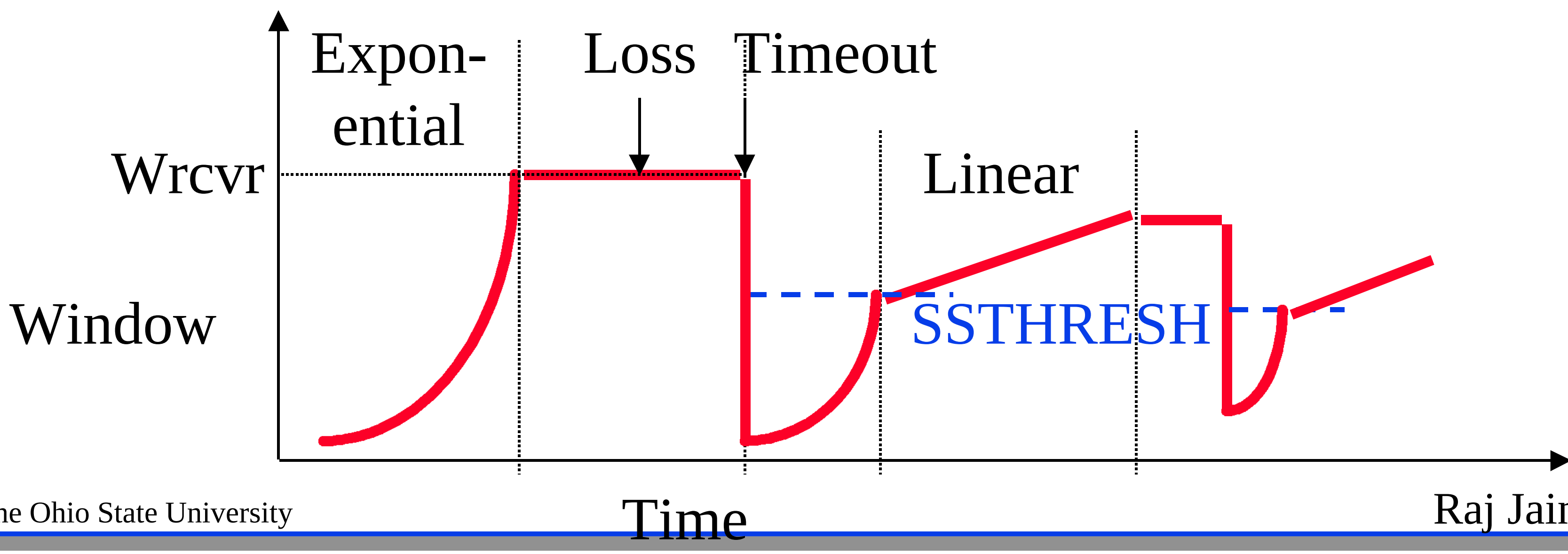
- TCP Congestion mechanisms
  Slow start, Fast Retransmit/recovery, RED, ECN
- TCP over ABR results
- Tough TCP Tests: Further modifications of ERICA
- TCP over UBR results
- UBR+

TCP Congestion Mechanisms

- End-System Based:
  - Silly Window Syndrome Avoidance
  - Delayed Ack
  - Slow Start Congestion Avoidance
  - Fast Retransmit and Recovery
  - *Selective Acknowledgment (SACK)*
- Router Based:
  - *Random Early Discard (RED)*
  - *Explicit Notification (Future)*
- Difference between routers and switches is decreasing
  It is important to understand TCP/IP mechanisms

TCP/IP Slow Start

- Maximum Segment Size (MSS) = 512 bytes
- Congestion Window (CWND)
- Window W = Min{Wrcvr, CWND}
- Slow-Start Threshold = max{2,min{CWND/2,Wrcvr}}
- Exponential until SSTHRESH: W = W+1 for every ack
- Linear afterwards: W = W + 1/W for every ack until Wrcvr



ABR with Small Buffers

# srcs	TBE	Buffer Size	T1	T2	T3	T4	T5	Through put	% of Max	CLR.
2	128	256	3.1	3.1				6.2	10.6	1.2
2	128	1024	10.5	4.1				14.6	24.9	2.0
2	512	1024	5.7	5.9				11.6	19.8	2.7
2	512	2048	8.0	8.0				16.0	27.4	1.0
5	128	640	1.5	1.4	3.0	1.6	1.6	9.1	15.6	4.8
5	128	1280	2.7	2.4	2.6	2.5	2.6	12.8	21.8	1.0
5	512	2560	4.0	4.0	4.0	3.9	4.1	19.9	34.1	0.3
5	512	5720	11.7	11.8	11.6	11.8	11.6	58.4	100.0	0.0

- CLR has high variance
- CLR does not reflect performance. Higher CLR does not necessarily mean lower throughput
- CLR and throughput are one order of magnitude apart

TCP over ABR: Observations

- CLR in the switch is low. But, throughput is also low
- The buffers can not be allocated based on TBE
- Maximum queue length and TBE have little/no relationship

TCP: Observations

- With enough buffers in the network, TCP can automatically fill any available capacity.
- TCP performs best when there is NO packet loss.
  Even a single packet loss can reduce throughput considerably.
- Slow start limits the packet loss but loses considerable time.
  With TCP, you may not lose too many packets but you loose time.
- Bursty losses cause more throughput degradation than isolated losses.
- With low buffers, TCP does not use all the available bandwidth
- Many duplicate packets are dropped at the destination

- For each packet loss, much time is lost due to timer granularity
  Timer granularity is the key parameter in determining time lost

Fast Retransmit and Recovery

- Fast Retransmit: Three consecutive acks for the same segment ⇒ Loss ⇒ Retransmit before timeout
- Fast Recovery: Reduce congestion window to half (instead of 1)
  ⇒ No new transmissions until duplicate acks arrive for the remaining half
- Single packet loss ⇒ One RTT wasted ⇒ Not bad
- Multiple packet loss ⇒ Timeout
  Timeout = Mean + 4 Stdv. of RTT or one tick
  ⇒ 100 ms wasted ⇒ Really bad

Effect of Fast Retransmit

- Fast retransmit helps only if occasional losses
  Mild congestion or errors
- With n packet loss, SSTHRESH is reduced to half after each retransmission. Window enters the linear-increase zone even when the window is small $\Rightarrow$ Low throughput.
- Even with fast retransmits, there are time-outs when the losses are bursty. These time-outs are more damaging than if there is no fast retransmit since SSTHRESH is low.

	Bursty Loss	Scattered Loss
With Fast-Retransmit Fast-Recovery	×	√
Without Fast-Retransmit Fast-Recovery	√	×

Buffer Requirements for ABR: Key Factors

- Switch Algorithm: Transient Response (settling) time
- Round Trip Time (RTT)
- Feedback Delay (bottleneck to source)
- Switch Algorithm *Parameters*:
  - Averaging Interval
  - Target Utilization
  - ERICA+ queue control
- Presence and characteristics of background VBR
- Number of VCs
- TCP Receiver window size

ABR Switch Buffer Requirements

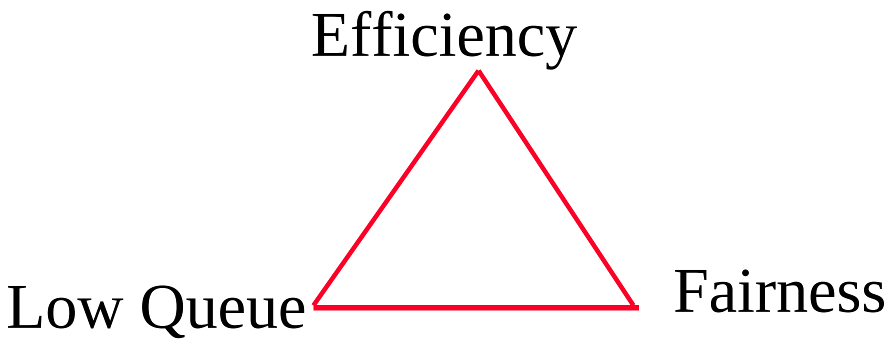
- ABR performance depends heavily upon the switch algorithm.
  Following statements are based on our *modified ERICA* switch algorithm.
  No cell loss for *TCP* if switch has Buffers = 4 × RTT.
- No loss for any number of TCP sources w 4 × RTT buffers.
- No loss even with VBR. W/o VBR, 3×RTT buffers will do.
- Under many circumstances, 1× RTT buffers may do.
- With ABR most of the queues are at the source.
  Not much queue in the switch
- In general:
  Qmax = a × RTT + b × Averaging Interval + c × Feedback delay + d × fn(VBR)

High Frequency VBR: Problem

- Limit of 1 × RTT due to VBR is good for large VBR cycle times.
  TCP and ABR get enough time to adjust.
- Faster VBR causes faster variations in available capacity.
  Neither TCP nor Switch algorithm may have time to adjust
  ⇒ Can lead to instability at high utilization levels.

VBR On/Off	RTT	F/b Delay	Maximum Queue	Total Throughput	Effici-ency	Fair-Ness
30 ms	30	10	12359=1.12*RTT	69.60	92.65	0.9967
100 ms	30	10	13073=1.18*RTT	63.85	85.00	0.9987
10 ms	30	10	diverges			
1 ms	30	10	diverges			

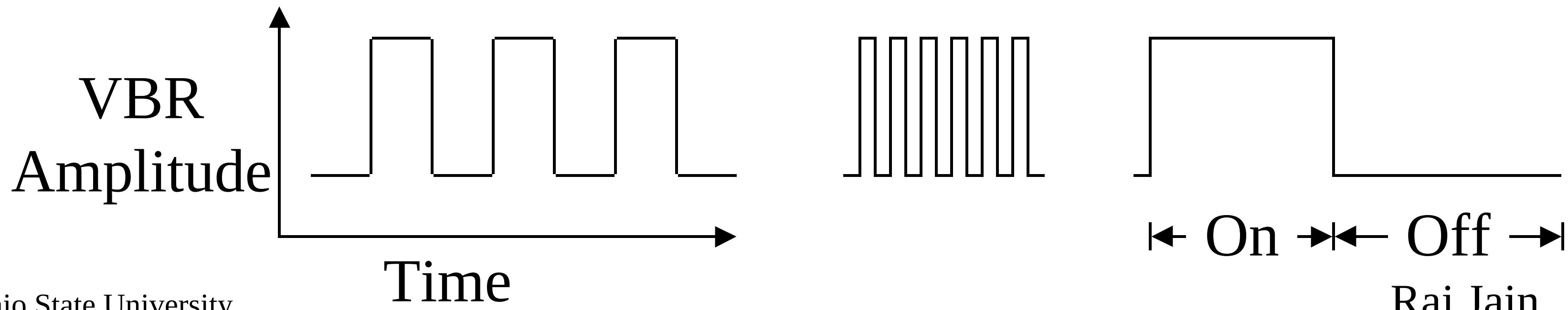
Three Way Tradeoff



- Buffers vs Efficiency (Utilization) vs Fairness
- It is possible to have lower queues (lower buffer required) if the target utilization is kept low.

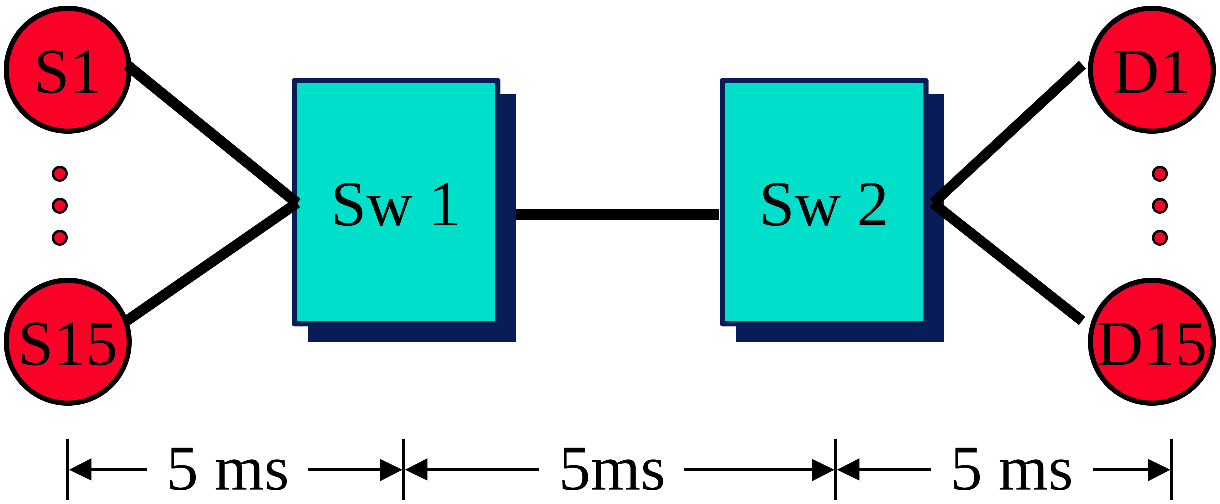
ABR Test Cases

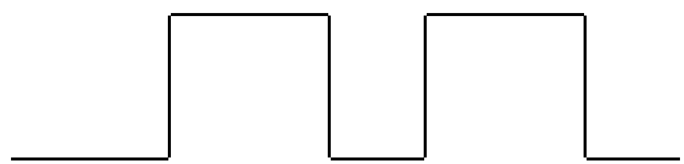
- Configurations: n sources, parking lot, ...
- ABR Traffic: Infinite, bursty
- Background traffic: without and with VBR
- High Layer: non-TCP, TCP
- RTT mix: Similar RTTs, varying RTTs
- VBR Period: Large, medium, small (compared to feedback delay)
- VBR Duty cycle: 0.9, 0.8, 0.7, ...



A Tough Test Case

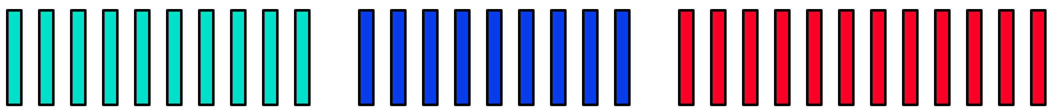
- 15 TCP sources, with 10 ms VBR period with a duty cycle of 0.7 (7 ms on, 3 ms off),
  10 ms feedback delay,
  30 ms RTT





Flocking Effect

- All cells of a VC are often seen together.
- There is clustering of sources.
- Not all sources are seen all the time.



ERICA Modifications

- Boundary Cases:
  - No ABR cells received
    $\Rightarrow$ No active sources (N=0)
    Fairshare = ∞?

ERICA Modifications (Cont)

- Average number of sources
- Average load factor = ABR Input rate/ABR capacity
  - Average ABR Input Rate
    = Number of cells/averaging interval
    - Average Number of ABR cells
    - Average Averaging interval
  - Average VBR usage
    - Average Number of VBR cells
    - Average Averaging interval
- Averaging ⇒ Decisions based on longer timer
  ⇒ Slower response
  ⇒ Buffer requirements are over 4 × RTT

TCP Over Plain UBR

- Low throughput
- Unfair
- Anomalies: More receiver buffer $\Rightarrow$ Lower throughput
  Due to Silly window avoidance + Delayed Ack
- Solution: Min sender buffer size should be $3 \times$ MSS

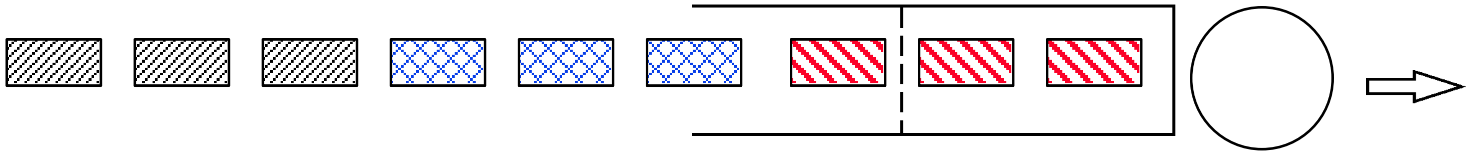
Ref: Comer

TCP/IP over UBR: Improvements

- Switch Based Mechanisms:
  - PPD
  - EPD
  - EPD + per-VC Accounting
  - EPD + per-VC queueing
- Source Based Mechanisms:
  - CLP Probe
  - Cell Pacing
  - Smaller Segments

PPD and EPD

- Plain ATM: Discard all cells if Q > threshold
- Partial Packet Discard:
  Discard all cells of a packet if one cell dropped
  Q > threshold
- Early Packet Discard:
  Discard all cells of new packets if Q > threshold



PPD vs EPD

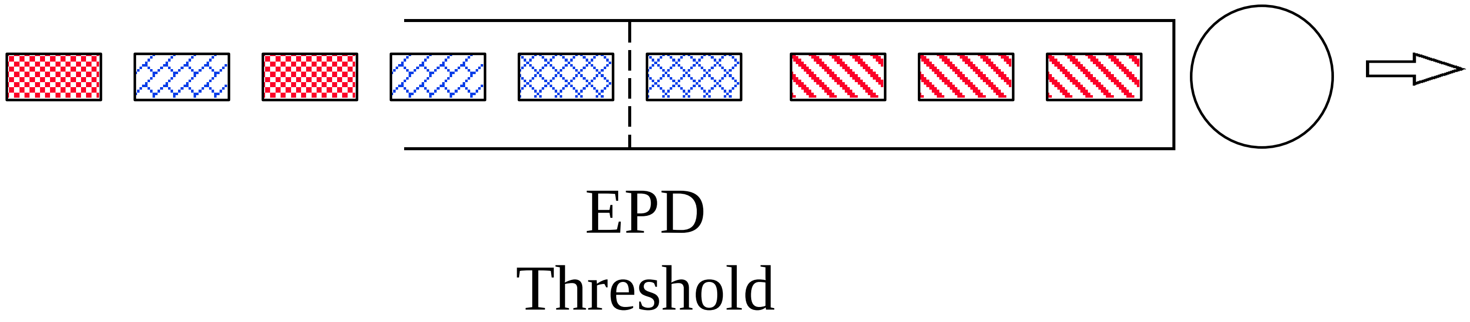
- Plain ATM ⇒ Many packets dropped
- Dropping all cells of a packet is better than dropping randomly
  ⇒ PPD is better than plain UBR
- Never drop the EOM cell of a packet.
  It results in two packet losses.
- EPD ⇒ Even fewer packets dropped => better throughput
- Plain ATM << PPD << EPD
- EPD improves efficiency but not fairness

UBR Switch Buffer Requirements

- Switch queues may be as high as the sum of TCP windows
  No cell loss for TCP if Buffers = Σ TCP receiver window
- Required buffering depends upon the number of sources.
- TCP receiver window ≥ RTT for full throughput with 1 source.
- Unfairness in many cases.
- Fairness can be improved by proper buffer allocation, drop policies, and scheduling.
- Drop policies are more critical (than ABR) for good throughput
- No starvation ⇒ Lower throughput shows up as increased file transfer times = Lower capacity

EPD + Per-VC Accounting

- Selective EPD: Select only high rate VCs
  Fast Buffer Allocation Scheme
- EPD: Drop all packets if queue X > threshold R ⇒ Unfair
- No per-VC queueing ⇒ All VCs share a single FIFO queue
- per-VC accounting ⇒ track Xi and N
- N = # of non-zero Xi's

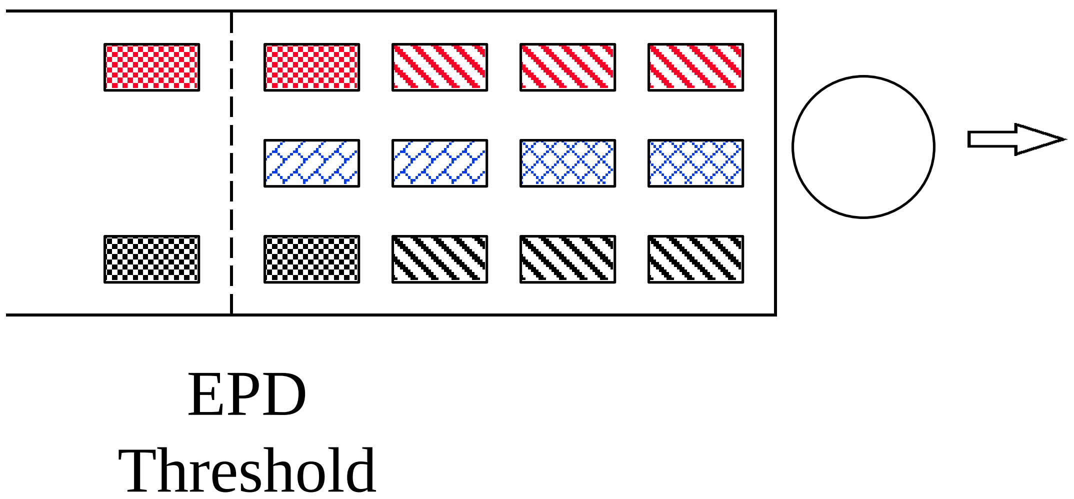


- If X > threshold, drop next arriving packet
  if Xi >= fn(X, N, K, R)
- Where K = Total number of buffers
- Drop if $Xi/(X/N) \geq Z(1+ (K-X)/(X-R))$
- Here Z = parameter between 0.5 and 1
- Note that packets from more and more flows are dropped as queue X increases
- Siu has analyzed a EPD + Simpler per-VC accounting
- If X > threshold, drop next arriving packet if $Xi/(X/N) \geq Z$
- **Conclusion:** Per-VC accounting improves fairness
- Other Ideas:
  - Do not drop successive packets
  - Drop from queues not tails ⇒ earlier effect

EPD + Per-VC Queueing

- Accept the next packet if Xi/(X/N) < Z
- Round-robin scheduling ⇒ Fairness further improved
- However, more VC's have packets dropped
  ⇒ Lower total throughput

Ref: Siu



CLP Probe

- Idea:
  - Use probe packets with CLP bit set to sense network congestion
  - If probe makes it then increase window.
  - otherwise slow-start
- Whenever window is increased, the next packet is sent with CLP set
- Throughput improved from 53% to 85%

Ref: Perloff and Reiss

Cell Pacing

- Use lower than link rate
- Using the right rate changed throughput from 0.9% to 68%
- Even with multiple sources throughput changed from 1.6% to 52%
- How to select the right rate?

Ref: Ewy, et al

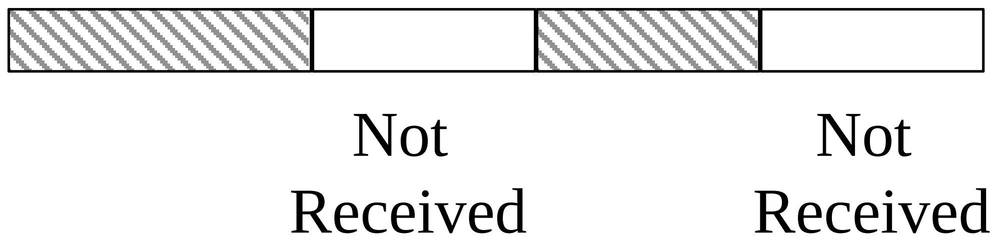
Effect of Segment Size

- Large segments ⇒ Large retransmissions
- If buffering is the bottleneck, smaller segments ⇒ better throughput
- If processing is the bottleneck, smaller segments ⇒ More overhead ⇒ Less throughput
- Buffering was small in initial switches
- Need buffering equal to several round-trips

Ref: Ewy, et al

Selective Acknowledgment

- Allows receiver to indicate multiple blocks of received segments.
- Receiver indicates lower-edge and upper-edge of all received segments
- Senders can retransmit only the missing segments.
- There is no need for multiple timeouts or duplicates



Random Early Discard

- Exponential averaging
  - Bursty traffic
    $\Rightarrow$ Instantaneous queue can be high or low
    $\Rightarrow$ time averaging
  - Qavg = $(1-\alpha)$Qavg+ α q if q>0
  - Qavg = $(1- \alpha)^{\beta}$ Qavg otherwise, β = f(idle time)
- Two thresholds:
  - Min < Qavg < Max $\Rightarrow$ mark (drop) arriving packet with probability p
  - pb = pb_max $\times$ (Qavg-Min)/(Max-min)
  - p = pb/(1-count $\times$ pb)

- Count = Number of packets since the last mark
  $\Rightarrow$ Marking probability increases as more packets arrive
- Count reset after marking a packet
- Qavg $\geq$ Max_threshold $\Rightarrow$ Drop all

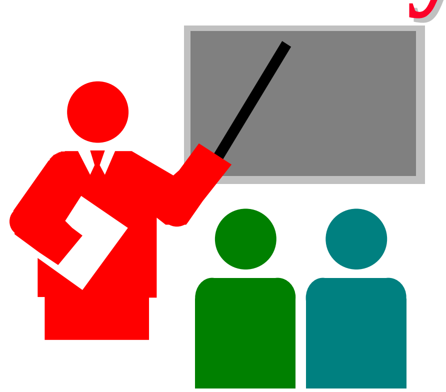
Ref: Floyd and Jacobson

Explicit Notification

- Routers send ICMP messages (or set bit) when Qavg > threshold
- Sources respond to ECN once per round trip $\Rightarrow$ Ignore others
- Halve cwnd and ssthresh on first ECN
- Do not respond to succeeding ECNs until all outstanding packets have been acked (i.e., one round trip)
- Do not reduce cwnd or ssthresh after timeout or 3 duplicate acks, if ECN action taken in this round trip.

Ref: S. Floyd, "TCP and Explicit Congestion Notification," LBL Tech Report.

Summary



- Packet loss results in a significant degradation in TCP throughput. For best throughput, TCP needs no loss.
- With enough buffers, ABR may guarantee zero loss for any number of TCP sources.
- Performance of ABR depends on the switch algorithm
- For zero loss, UBR need buffers = Σ receiver windows
- PPD << EPD << Selective EPD
- ABR vs UBR issue is that of ATM end-to-end vs backbone

TCP Congestion: References

- W. Richard Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms," Internet Draft draft-stevens-tcpca-spec-00.txt, February 1996
- V. Jacobson and M. Karels, "Congestion Avoidance and Control," Proc. SIGCOMM'88, August 1988, pp. 314-329.
- S. Floyd and V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance," IEEE/ACM Transactions on Networking, Vol. 1, August 1993, pp. 397-413.
- J. Nagle, "Congestion Control in IP/TCP Internetworks," ACM Computer Communications Review, Vol. 14, No. 4, October 1984, pp. 11-17.

- V. Jacobson, R. Braden, and D. Borman, "TCP Extensions for High Performance," Internet RFC 1323, May 1992.
- S. Floyd, "TCP and Explicit Congestion Notification,"
- S. Floyd, "TCP and Successive Fast Retransmits," LBL Technical Report, May 1995, available ftp:://ftp.ee.lbl.gov/papers/fastretrans.ps
- C. Villamizar and C. Song, "High Performance TCP in ANSNET," Computer Communications Review, October 1994, pp. 45-60.
- J. Hoe, "Improving the start-up behavior of a congestion control scheme for TCP," ACM SIGCOMM 96, to appear.
- L. Zhang, S. Shenker, and D.D. Clark, "Observations on the dynamics of a congestion control algorithm: the effects of two-way traffic," SIGCOMM'91, September 1991, pp. 133-147.

TCP over ATM: Our Papers/Contributions

All our past ATM forum contributions, papers and presentations can be obtained on-line at http://www.cis.ohio-state.edu/~jain/

- S. Kalyanaraman, R. Jain, S. Fahmy, R. Goyal, F. Lu and S. Srinidhi, ``Performance of TCP/IP over ABR," To appear Globecom'96.
- R. Jain, et al, "Buffer Requirements for TCP over ABR," ATM Forum/96-0517, April 1996.
- R. Jain, et al, "Performance of TCP over UBR and buffer requirements," ATM Forum/96-0518, April 1996.
- R. Jain, et al, "TBE and TCP/IP traffic," ATM Forum/96-0177, February 1996.

TCP over ATM: References

- A. Romanow and S. Floyd, "Dynamics of TCP Traffic over ATM Networks," IEEE Journal on Selected Areas in Communications, Vol. 13, No. 4, May 1995, pp. 633-641, ftp://ftp.ee.lbl.gov/papers/tcp_atm.ps.Z
- J. Heinanen and K. Kilkki, "A Fair Buffer Allocation Scheme," Telecom Finland Draft 17 March 1995.
- H. Li, K-Y Siu, and H-Y Tzeng, "TCP Performance over ABR and UBR Services in ATM," Proc. IPCCC'96, March 1996.
- D. E. Comer and J. C. Lin, "TCP Buffering and Performance over an ATM Network," Internetworking: Research and Experience, Vol. 6, 1995, pp. 1-13.

- M. Perloff and K. Reiss, "Improvements to TCP Performance in High-Speed ATM Networks," Communications of ACM, February 1995, pp. 90-100.
- B.j. Ewy, et al, "TCP/ATM Experiences in the MAGIC Testbed,"
- L. Kalampoukas and A. Varma, "Performance of TCP over Multi-Hop ATM Networks: A Comparative Study of ATM-Layer Congestion Control Schemes," Technical Report, UCSC-CRL-95-13, in ftp://ftp.cse.ucsc.edu/