# TCP Congestion Control with Appropriate Byte Counting

**`draft-allman-tcp-abc-00.txt`**

Mark Allman, NASA GRC/BBN

TSV WG Meeting

August 2000

# Introduction

- [RFC2581] specifies the way a TCP increases *cwnd*:
  - Slow Start: By at most 1 MSS for each ACK
  - Congestion Avoidance: By at most 1 MSS each RTT
- The ABC draft is intended to offer additional guidance on increasing *cwnd*
  - More appropriate *cwnd* increase
  - Provides protection against misbehaving receivers
  - Provides performance benefits (maybe)

# Appropriate Byte Counting

- The idea behind Appropriate Byte Counting (ABC) is to increase *cwnd* based on the number of new bytes ACKed by each arriving acknowledgment, rather than based on the number of ACKs that arrive.

- Denote the number of new bytes ACKed by each arriving acknowledgment by $NBA$

# Appropriate Byte Counting (cont.)

- Congestion Avoidance:
  - Upon each ACK arrival increase $bytes\_acked$ by $NBA$
  - If $bytes\_acked \geq cwnd$:
    - $bytes\_acked -= cwnd$
    - $cwnd += 1\ MSS$
  - Note: This is explicitly allowed by [RFC2581].
- Slow Start:
  - Upon each ACK arrival, increase *cwnd* by $min(L, NBA)$ for some limit $L$ bytes

# Appropriate Byte Counting (cont.)

- Choosing $L$:
  - If $L = 1\ MSS$ ABC is *no more* aggressive than allowed by [RFC2581] and sometimes less aggressive.
  - If $L > 1\ MSS$ ABC is *more* aggressive than currently allowed.
    - This can lead to large line-rate bursts of data in response to "stretch ACKs" (caused by non-compliant delayed ACK implementations or ACK loss).

# Appropriate Byte Counting (cont.)

- The document suggests that $L = 2\ MSS$ to balance between being very conservative and being too aggressive.
  - $L = 2\ MSS$ exactly negates the *cwnd* growth sacrificed by delayed ACKs.
- The exception is in slow start following an RTO when a TCP must use $L = 1\ MSS$
  - Incoming ACKs are often acknowledging a bunch of previously unacknowledged data that has been sitting in receive buffers – this says nothing about the congestion state of the network.

# Advantages

- More appropriate *cwnd* increase [All98]:
  - If an ACK covers only a few bytes, why give it many times that much permission to send?
- Mitigates the performance impact of delayed ACKs without additional network traffic [All98,All99]:
  - *cwnd* increase based on the number of arriving ACKs coupled with delayed ACKs increases the time required to increase *cwnd* by roughly a factor of 2.
  - ABC can avoid this by increasing *cwnd* more for each delayed ACK (and holding the increase for an ACK of a single segment constant).
  - Especially important for short transfers.

# Advantages (cont.)

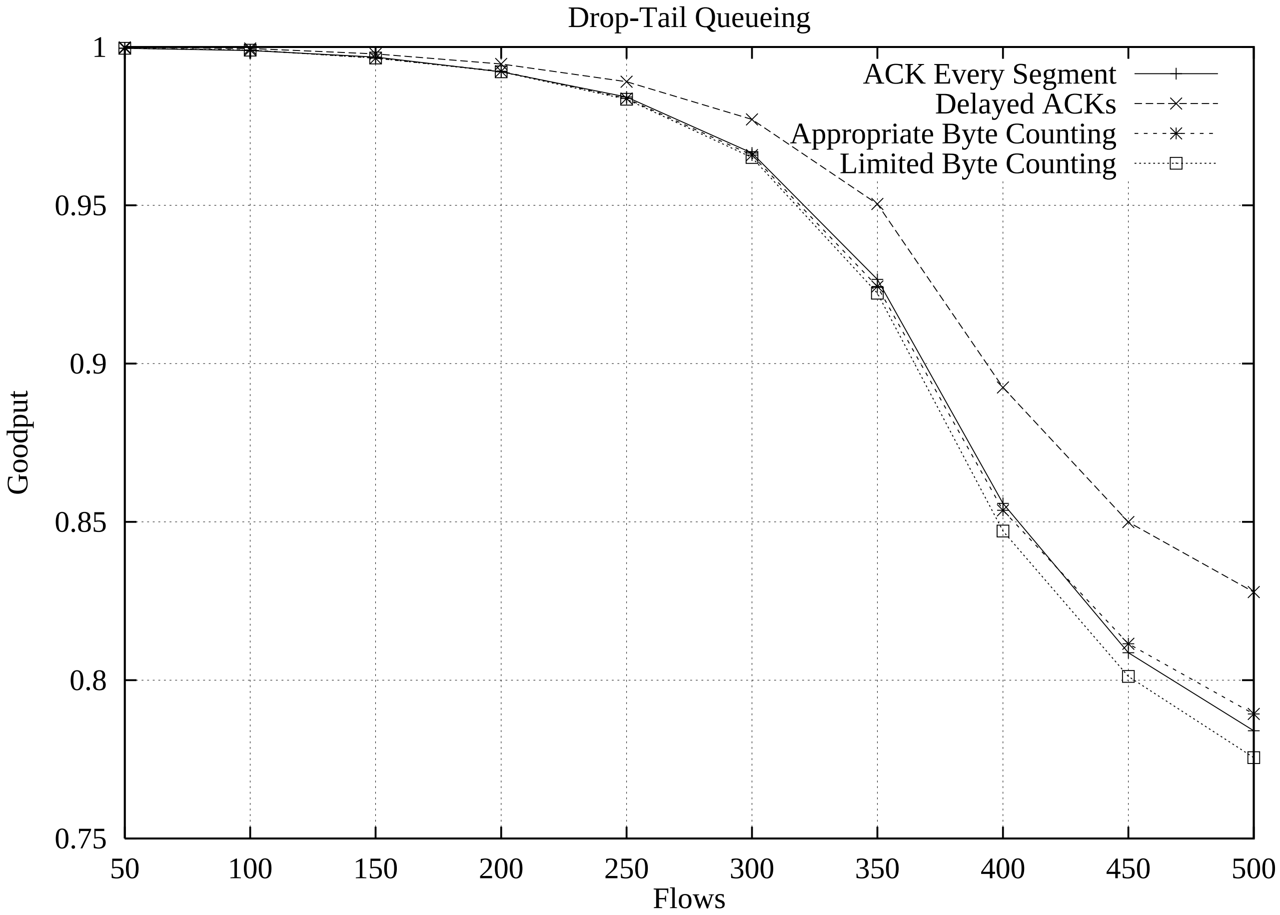
- Prevents attacks from misbehaving receivers [SCWA99]:
  - A receiver can use "ACK splitting" to induce the sender into inappropriately increasing *cwnd* (because the increase is tied to the number of ACKs that arrive).

## Disadvantages

- Assuming $L = 2\ MSS$, increased burstiness:
  - Micro-burstiness increase: The number of data segments sent in response to an ACK is increased by 1 segment.
  - Macro-burstiness increase: During slow start *cwnd* is currently increased by roughly 50% per RTT (assuming delayed ACKs) whereas ABC will cause an increase of roughly 100% per RTT.
- If $L = 1\ MSS$ the disadvantages go away.

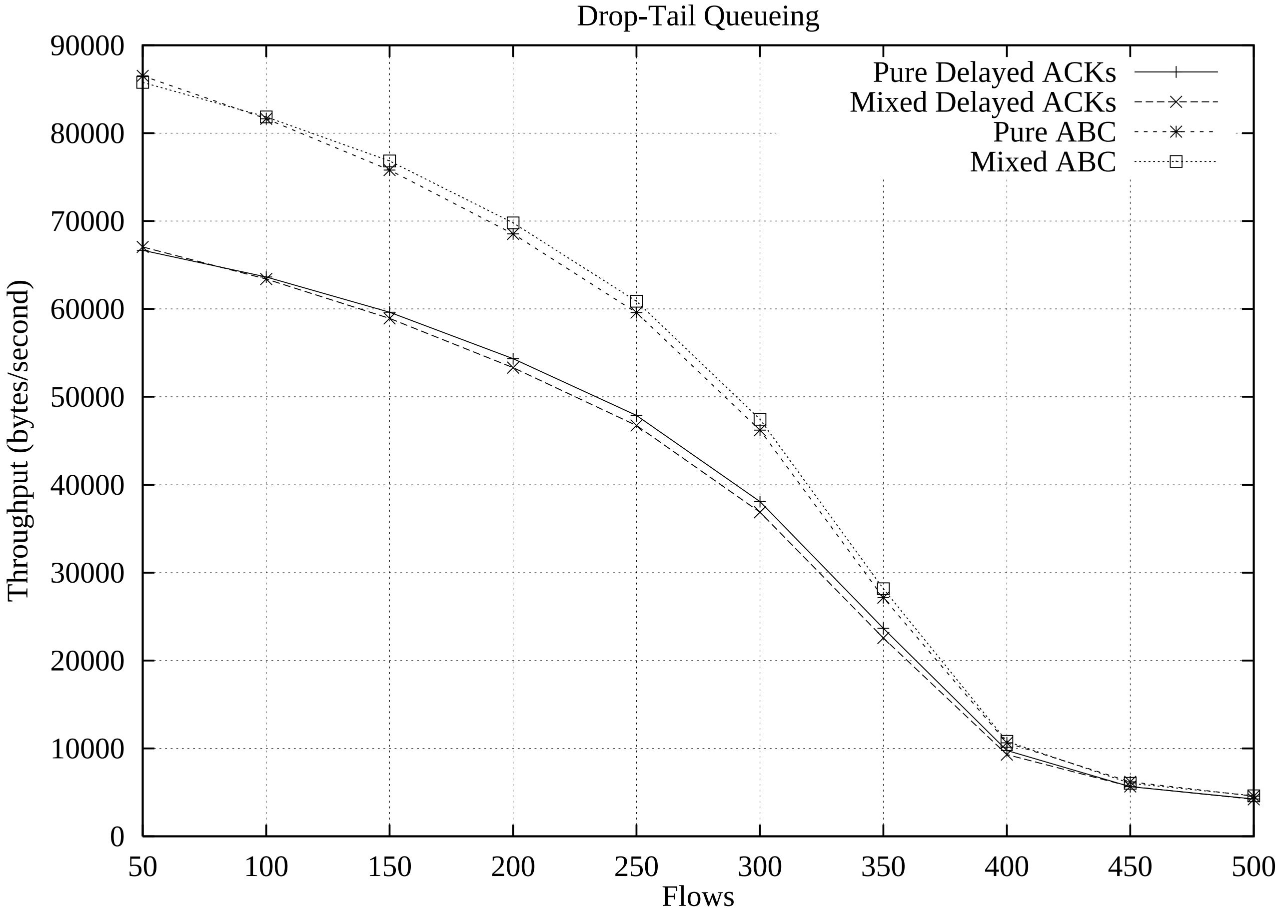
# Disadvantages (cont.)

- Illustration of increased loss rate:



# Fairness

- Simulations indicate the ABC, with $L = 2\ MSS$ is not unfair to competing traffic (whether using ABC or not).



# Conclusions

- ABC, with $L = 1\ MSS$, should be used.
- ABC, with $L = 2\ MSS$, is likely OK for general purpose use and we should encourage widespread testing of the mechanism in the Internet.

# References


- [All98] Mark Allman. On the Generation and Use of TCP Acknowledgments. ACM Computer Communication Review, 28(5), October 1998.
- [All99] Mark Allman. TCP Byte Counting Refinements. ACM Computer Communication Review, 29(3), July 1999.
- [RFC2581] Mark Allman, Vern Paxson, W. Richard Stevens. TCP Congestion Control, April 1999. RFC 2581.
- [SCWA99] Stefan Savage, Neal Cardwell, David Wetherall, Tom Anderson. TCP Congestion Control with a Misbehaving Receiver. ACM Computer Communication Review, 29(5), October 1999.