Early Retransmit for TCP

draft-allman-tcp-early-rexmt-00.txt

Mark Allman, Konstantin Avrachenkov, Urtzi Ayesta, Josh Blanton

TSVWG - IETF 56 - March 2003

Motivation

- There are cases when TCP cannot use fast retransmit to recover from loss which requires the use of the (often costly) RTO.
- E.g., consider a TCP connection with a cwnd of 3 packets, 1 of which gets dropped and nothing new to send
  - there is no chance of getting 3 duplicate ACKs
- Limited Transmit (RFC 3042) helps if we can send new data
  - But, sometimes the app has nothing else to send
  - And, sometimes the receiver's advertised window will not allow the sender to transmit new packets

Early Retransmit

- If the cwnd < 4*MSS and the sender cannot send new data ...
- The the sender can trigger fast retransmit on cwnd - 1 duplicate ACKs

History

- Originally, ER was packaged with Limited Transmit, but was not as enthusiastically embraced as LT
  - so, we split the two to get LT done

Costs and Benefits

- Benefits
  - avoid RTOs in cases where we cannot currently trigger fast retransmit

- Costs
  - less robust to reordering

Worst Case #1

- Tons of connections that ...
  - are 2 data packets long
  - have the data two packets reordered
- In this case each connection will send a spurious retransmit (so, one-third of the data packets will be useless)

Worst Case #2

- A long connection...
  - the app generates two packets of data at a time
  - the two packets are reordered
- In this case, we get a long connection that continually sends 1 spurious retransmit for every 2 useful data packets.

Open Questions

- So, are the benefits worth the costs?
  - are the worst cases sufficiently rare to ignore them?
- Should we add some sort of bound on the number or periodicity of ERs?
  - If so, what should that limit be?