TCP User Timeout Option

Lars Eggert (NEC)
Fernando Gont (UTN/FRH)
IETF-64, Vancouver, Canada

November 10, 2005
Problem

- TCP closes a connection if transmitted data has not been ACK’ed after a specific period of time ("user timeout")
- default duration (RFC793 & 1122) is minutes
- behavior makes it difficult for connections to survive extended connectivity disruptions
  - if they are transmitting when a disruption occurs or begin to transmit during one
User Timeout

• per-connection, app-controlled setting that specifies how long TCP will retransmit & wait for ACKs of un-ACK’ed data

• parameter to RFC793 SEND call

• often implemented differently, e.g., BSD SO_SNDBUF sockopt
Issue

- apps need to coordinate setting their local UTOs for a connection
- even if app on one peer lengthens the UTO of a connection, it can still close if the peer doesn’t and attempts to transmit during a connectivity disruption
- modifying all apps for disruption tolerance is possible but tedious
UTO Option

- advisory TCP option that signals local initial UTO & app-initiated changes to it to the peer’s TCP stack
- if the peer app has not explicitly set its UTO, peer stack SHOULD adapt its local UTO based on received UTO
- eliminates the need for app changes at both ends of a connection
Changes Since -01

• app-specified UTO takes precedence over received UTO option (T. Faber)
• discuss SO_SNDTIMEO & SO_RCVTIMEO
• wording & terminology (J. Touch)
• some shuffling of sections
Are We Done?

• no other (technical) changes pending
• have received no comments on -02
• can this version move forward?