TCP ACK Rate Request (TARR) option

draft-gomez-tcpm-ack-rate-request-03

Carles Gomez
Universitat Politècnica de Catalunya

Jon Crowcroft
University of Cambridge

IETF 113 Vienna, TCPM WG, March 2022
Motivation

• Delayed ACKs
  • Intended to reduce protocol overhead
  • But may also contribute to suboptimal performance

• “Large” cwnd scenarios (i.e. cwnd $\gg$ MSS):
  – Saving up to 1 of every 2 ACKs may be insufficient
    • Performance limitations due to asymmetric path capacity
    • Computational cost and network load

• “Small” cwnd scenarios (i.e. cwnd up to $\sim 1$ MSS):
  – Data centers: BDP up to $\sim 1$ MSS
    • Delayed ACKs will incur a delay much greater than the RTT
  – Transactional data exchanges, or when cwnd decreases
    • Immediate ACKs may avoid idle times, allow faster cwnd growth
Status

• Related prior discussion
  • Sender control of TCP ACKs
  • Converged to defining a new TCP option serving two purposes:
    – Requesting a given ACK rate
    – Requesting immediate ACKs

• Version -03
  • Aims to address comments from (many thanks!):  
    – Yoshifumi Nishida
    – Michael Scharf
    – Jonathan Morton
    – Bob Briscoe
Updates in -03 (I/V)

• Main format
  • OLD:
    - When R=0, sender requests immediate ACKs for the next N segments
    - However, mostly redundant
  • NEW:
Updates in -03 (II/V)

• Two possible encodings for the R field:

  • OPTION 1:
    – Binary encoding of the requested ACK rate
    – The maximum value of R is 63

  • OPTION 2:
    – 4 leftmost bits represent a mantissa (m)
    – 2 rightmost bits represent an exponent (e)
    – The requested ACK rate is \( R = (m+1) \times 2^{2 \times e} \)
    – The maximum value of R is 1024
Updates in -03 (III/V)

• Section 3:
  • A TARR-option-capable receiving TCP SHOULD modify its ACK rate to one ACK every R received data segments from the sender
    – Reasons why not a MUST: lack of resources, security...
    – R=1: the receiving TCP SHOULD send an ACK immediately
    – R=0: not defined
  • Upon reception of a SYN carrying the TARR option, a TARR-option-capable endpoint MUST include the TARR option in the SYN-ACK sent in response
    – Question: due to lack of SYN space, including TARR only in response to the SYN-ACK?
  • A TCP segment carrying retransmitted data is not required to include a TARR option
  • Question: is the Ignore Order feature considered useful?
    – If RACK not supported, long loss detection time
Updates in -03 (IV/V)

• New section 5: changing the ACK rate during the lifetime of a TCP connection
  • ACK rate may depend on cwnd (may change during a connection)
    – cwnd should settle in congestion-avoidance phase
    – Routing, path capacity, path load changes may impact the BDP (thus cwnd and the ACK rate)
  • Ability to suppress DelACKs to allow measuring the RTT for each packet in some intervals; allow different ACK rate afterwards
  • Linux receiver heuristic to detect slow start and suppress Delayed ACKs until its end
    – Some slow start variants may confuse the heuristics.
    – Explicit end of slow start signal may be useful to avoid slow start sender behavior ossification
  • Reducing ACK load when ACK decimation is detected by the sender
Updates in -03 (V/V)

• Security considerations
  • TCP-AO may be used to protect TCP segment header
  • Guidance and attack mitigation given in RFC 5961 is RECOMMENDED for a TARR receiver
    – TARR option MUST be ignored on a packet deemed invalid
  • A TARR receiver might opt not to fulfill a request to avoid or mitigate an attack:
    – A large number of senders requesting immediate ACKs simultaneously after a large number of data segments sent
Next steps

• Continue improving the document
  • Further feedback will be appreciated

• Looking for collaboration
  • Implementation
    – Running code
  • Co-authorship
Thanks!
Questions? Comments?

Carles Gomez
Universitat Politècnica de Catalunya

Jon Crowcroft
University of Cambridge

IETF 113 Vienna, TCPM WG, March 2022