TCP ACK Rate Request (TARR) option

draft-ietf-tcpm-ack-rate-request-01

Carles Gomez
Universitat Politècnica de Catalunya

Jon Crowcroft
University of Cambridge

IETF 116 Yokohama, TCPM WG, March 2023
Intro: motivation

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

• “Large” cwnd scenarios (i.e. cwnd >> 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 ~1 MSS):
  – Data centers: BDP up to ~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
Intro: main TARR option format

- R carries binary encoding of ACK rate
- Maximum value of R: 127

“R” is the requested ACK rate
- R = 0: request an immediate ACK

Reserved bit (e.g. for future encodings/values of R, if needed)
Status

• WG adoption
  • draft-ietf-tcpm-ack-rate-request-00
    – Same content as draft-gomez-tcpm-ack-rate-request-06
  • February 2023

• Version -01
  • Aims to address comments received on the mailing list
    – Stretch ACKs
  • Minor additions and corrections
Updates (I/VI)

• Section 3 additions

• 3.1. Sender behavior
  – A TCP sender MUST NOT communicate a value of R corresponding to an amount of data bytes to be acknowledged at once by the receiver greater than rwin size or greater than cwnd size

• 3.2. Receiver behavior
  – A TARR-option-capable receiving TCP MUST ignore a value of R corresponding to an amount of data bytes to be acknowledged at once greater than the last rwin size it has announced
Updates (II/VI)

• New Section 5: “Issues of Stretch ACKs”
  • TARR may produce Stretch ACKs
    – ACKs that acknowledge more than two previously unacknowledged data segments
  • Issues:
    – Sender burstiness
    – Slow cwnd opening
    – Lower frequency of RTT samples
Updates (III/VI)

• 5.1. Sender burstiness
  • May contribute to router queue overflow and packet loss
  • Possible mitigation:
    – TCP Sender Pacing
      » Requires an algorithm to determine the data segment transmission rate, commensurate with R
Updates (IV/VI)

5.2. Slow cwnd opening

- Problem
  - During slow start, cwnd increases by up to SMSS upon receipt of an ACK covering new data
  - Stretch ACKs (even Delayed ACKs) reduce the amount of ACKs received by the sender
    » Reduced rate of cwnd growth, increased transfer time, reduced throughput
  - ABC (RFC 3465) might solve the problem, but still experimental, not fully included in RFC 5681

- Solution
  - A TCP sender SHOULD NOT use TARR to produce Stretch ACKs during Slow Start
  - A TCP sender MAY use TARR (R=1) for data segments transmitted during Slow Start
Updates (V/VI)

• 5.3. Lower frequency of RTT samples
  • Stretch ACKs reduce the number of RTT samples
    – Reduces responsiveness to RTT changes
  • Time-based packet loss detection becomes inaccurate
    – Unnecessary delays and/or spurious retransmissions
  • A sender SHOULD trigger an ACK being sent by the receiver at least once per RTT. Options:
    – Sending a data segment with the TARR option with R=0 at least once per RTT
    – Using R > 0, producing at least one ACK per RTT
Updates (VI/VI)

• Minor additions
  • Section 1
    – Delayed ACKs “SHOULD” in RFC 1122
      » Added: “subsequently reinforced in RFC 5681”
    – Added reference to RFC 9006:
      » “TCP Usage Guidance in the Internet of Things (IoT)”
  • Section 4
    – OLD: “packets that do not have the SYN bit set”
    – NEW: “when the sender requests an ACK rate of R”
Thanks!

Questions? Comments?

Carles Gomez
Universitat Politècnica de Catalunya

Jon Crowcroft
University of Cambridge

IETF 116 Yokohama, TCPM WG, March 2023