TCP over Constrained-Node Networks

draft-gomez-core-tcp-constrained-node-networks-00

Carles Gomez
Universitat Politècnica de Catalunya (UPC)/Fundació i2cat
carlesgo@entel.upc.edu
Jon Crowcroft
University of Cambridge
Jon.Crowcroft@cl.cam.ac.uk

With the support of Ministerio de Educación, Cultura y Deporte, through the José Castillejo grant CAS15/00336
Motivation

• Several application layer protocols being used for the Internet of Things (IoT)
  – Constrained Application Protocol (CoAP)
    • Originally over UDP
    • CoAP over TCP in progress
      – To overcome middlebox problems
  – HTTP/2 and HTTP/1.1
  – MQTT
• TCP is being / will be used in many IoT scenarios
  – However, it has not received attention yet...
Main goal

• To offer simple measures to allow for lightweight TCP implementation and suitable operation in CNNs
Related WGs

- **CoRE**
  - CoAP, related framework
- **TCPM**
  - TCP maintenance and minor extensions
- **LWIG**
  - Lightweight implementation guidance
  - Suggested as the *home* for this draft
  - Not yet confirmed...
CNN characteristics

• Constrained nodes [RFC 7228]:
  – Significant limitations on
    • Processing, memory
    • Energy resources
  – Use *lossy* physical/link layer technologies
    • Wireless
    • Wired (but harsh, e.g. PLC)
  – Network topology
    • Star (single-hop)
    • Mesh (multihop)
TCP over CNNs

• Maximum Segment Size (MSS)
  – IPv6 requires support for 1280-byte packets
  – Many link layers have a short MTU
    • Tens to a few hundred bytes
  – 6Lo(WPAN) adaptation layers generally do not ensure support of IPv6 packet size > 1280 bytes
  – Therefore:
    • TCP MSS MUST NOT be set to > 1220 bytes
    • TCP MSS MUST NOT lead to IPv6 datagram size exceeding 1280 bytes
TCP over CNNs

• **Window Size**
  – Stop-and-wait (window size of one MSS)
    • Equivalent to CoAP end-to-end reliable mechanism
  – TCP often criticized as *too complex*, comments in CoRE WG to avoid reproducing TCP in CoAP
  – Stop-and-wait seems to be accepted for CoAP

• **For -01**
  – Recommend, not mandate, stop-and-wait
  – How to enable stop-and-wait operation
TCP over CNNs

• RTO estimation
  – CoCoA RTO SHOULD be used in TCP over CNNs
    • draft-bormann-core-cocoa

      \[
      \text{RTO} := 0.25 \times \text{E}_\text{weak} + 0.75 \times \text{RTO} \quad (1)
      \]

      \[
      \text{RTO} := 0.5 \times \text{E}_\text{strong} + 0.5 \times \text{RTO} \quad (2)
      \]

• Designed specifically for IoT scenarios
  – Adaptive RTO (based on RFC 6298), uses weak RTTs, Variable Backoff Factor, aging mechanism, dithering
  – Good PDR, settling time after a burst of messages, fairness

  – RFC 6298 RTO MAY be used
TCP over CNNs

• Keep-alive, TCP connection lifetime
  – TCP connection SHOULD be kept open if data will be sent (in the next two hours)
  – Keep-alive messages MAY be supported by a server
    • Useful to clean inactive connections state
    • Keep-alive timer cannot be set to less than 2 hours
      – Does not guarantee avoiding middlebox problems
      – Alternatives: frequent TCP connection establishment, application layer heartbeat messages

• For -01
  – Consider TCP Fast Open (RFC 7413)
  – Consider that many middleboxes fail to meet the recommended timeout of 124 min
TCP over CNNs

• Explicit Congestion Notification
  – ECN MAY be used in CNNs
  – When congestion signal reaches the sender and the sender window is of one segment
    • Rate reduced from $1/\text{RTT}$ to $1/\text{RTO\_default}$
  – Congestion control can be triggered earlier than upon reception of 3 duplicate ACKs or RTO expiration
TCP over CNNs

- TCP options
  - Stop-and-wait, therefore MUST NOT support
    - Window scale
    - TCP timestamps
    - SACK
- For -01
  - Parsing options 0, 1, and 2. Ignore options not wanted...
  - If not stop-and-wait, consider more options (e.g. SACK)
TCP over CNNs

• Explicit Loss Notifications
  – Would be useful to avoid activation of congestion control for corruption-induced losses
    • Lossy links in CNNs
  – Remains as experimental work
  – Not widely deployed
  – Not standardized by the IETF
Further items (for -01)

• Clarify scenarios
  – E.g. constrained device to unconstrained device

• Delayed ACKs

• Collect feedback from experiences with TCP in CNNs
  – What went wrong?
  – What went right?

• More flexibility, possibly remove (part of the) RFC 2119 language...
Thanks a lot for the feedback so far!

- Carsten Bormann, Zhen Cao, Wei Genyu, Michael Scharf, Ari Keranen, Abhijan Bhattacharyya, Andrés Arcia-Moret, Yoshifumi Nishida, Joe Touch, Fred Baker