Status

• Initial version presented in IETF 96
  – TCPM and LWIG

• Presented since then at LWIG
  – IETF 97, 98, 99, 100, 101, 102 (103 planned)

• Became an LWIG WG document after IETF 99

• TCPM WG has been in the loop through the ML
  – Heads-up given in IETF 102

• Last revision is -04
  – WGLC request
Section 1. Introduction

• TCP has often been criticized for IoT
  – Some of the reasons not valid
  – Consequence: TCP has sometimes been neglected for IoT
• However, TCP is being used in IoT
  – HTTP, CoAP (over TCP, now available), MQTT, etc.
• Goal of the document
  – Providing guidance on how TCP can be used/configured/implemented in IoT scenarios
Section 3. Characteristics of CNNs relevant for TCP

• 3.1. Network and link properties
  – Constrained devices: processing, memory, energy
  – Links: low bit rate, high loss rate, variable link quality
  – Multihop topology in many scenarios

• 3.2 and 3.3. Usage scenarios and traffic patterns
  – Unidirectional transfers, request-response, bulk
Section 4. TCP implementation and configuration in CNNs

• Organization

4. TCP implementation and configuration in CNNs ........................................ 6
   4.1. Path properties ................................................................. 6
      4.1.1. Maximum Segment Size (MSS) .................................... 7
      4.1.2. Explicit Congestion Notification (ECN) ......................... 7
      4.1.3. Explicit loss notifications ........................................... 8
   4.2. TCP guidance for small windows and buffers .............................. 8
      4.2.1. Single-MSS stacks - benefits and issues ....................... 8
      4.2.2. TCP options for single-MSS stacks ............................... 9
      4.2.3. Delayed Acknowledgments for single-MSS stacks ............ 9
      4.2.4. RTO estimation for single-MSS stacks ........................... 10
   4.3. General recommendations for TCP in CNNs ............................... 10
      4.3.1. Error recovery and congestion/flow control ................. 10
      4.3.2. Selective Acknowledgments (SACK) ............................. 11
      4.3.3. Delayed Acknowledgments ........................................... 11
Section 4.1. Path properties

• MSS
  – An adaptation layer (including fragmentation) required to support IPv6 over some link layers
    • Define an MTU of 1280 bytes
  – Other links support larger MTUs
  – Generally desirable to limit the MTU to 1280 bytes
    • Set the MSS not larger than 1220 bytes

• ECN
  – Beneficial in CNNs: reduce packet losses, sometimes only detected after an RTO expiration

• ELN
  – Mostly experimental, not standardized, not widely deployed
Section 4.2. TCP guidance for small windows

• Single-MSS send and receive window
  – Very simple congestion and flow control
  – Often sufficient for IoT (e.g. CoAP)
    • Not so good for bulk transfers
  – Many TCP options not required
    • Window scale, timestamps, SACK...

• Delayed ACKs
  – A single-MSS receiver would add unnecessary delay
  – A single-MSS sender would suffer unnecessary delay
    • Workaround: „split hack“ avoids delay (but adds overhead)

• RTO algorithm
  – Larger impact on performance if small sender window size
  – Tuning may be considered
Section 4.3. General recommendations for TCP in CNNs

- Fast Retransmit and Fast Recovery
  - Require large enough window size (e.g. 5 MSS)

- SACK
  - May avoid unnecessary retransmissions
  - Save energy, bandwidth; reduce latency

- Delayed ACKs
  - For small messages (< 1 MSS) or request/response: disabling delayed ACKs recommended (if possible)
  - For bulk transfers, delayed ACKs reduces number of ACKs
Section 5. TCP usage recommendations in CNNs

• **5.1. TCP connection initiation**
  – Typically, initiated by the constrained device

• **5.2. Number of concurrent connections**
  – Being conservative, recommended
    • RAM consumption, harmful in congested networks

• **5.3. TCP connection lifetime**
  – Long TCP connection desirable
    • Not always possible: middleboxes (e.g. firewalls)
  – Alternative: short TCP connections
    • Message overhead
  – TCP Fast Open (TFO)
    • Application-level idempotency not always guaranteed
  – Other alternatives
    • Application-layer heartbeats
    • TCP Keep-alives (if used, Keep-alive interval trade-off)
Section 6. Security considerations

- Best current practice applies
  - E.g. Use of TLS if applicable

- TCP security options
  - TCP MD5 signature, TCP-AO
    - Additional 18 bytes, 16-20 bytes (respectively)

- Shrew DoS attacks
  - One or more sources generate packets to coincide with consecutive retry attempts of a victim node, triggered by RTO expiration
  - Small window size senders are potential victims
  - Mitigation
    - RTO randomization
    - Attack blocking by routers, based on traffic patterns
## Annex. TCP implementations for IoT

<table>
<thead>
<tr>
<th>Feature</th>
<th>uIP</th>
<th>lwIP orig</th>
<th>lwIP 2.0</th>
<th>RIOT</th>
<th>TinyOS</th>
<th>FreeRTOS</th>
<th>uC/OS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code size (kB)</td>
<td></td>
<td>(&lt;5)</td>
<td>(~9)</td>
<td>(~14)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(a)</td>
<td></td>
<td>(T1)</td>
<td>(T2)</td>
<td>(T3)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Win size (MSS)</td>
<td></td>
<td>1</td>
<td>Mult.</td>
<td>Mult.</td>
<td>1</td>
<td>Mult.</td>
<td>Mult.</td>
</tr>
<tr>
<td>Slow start</td>
<td></td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Fast rec/retx</td>
<td></td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Keep-alive</td>
<td></td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Win. Scale</td>
<td></td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>TCP timestamp</td>
<td></td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>SACK</td>
<td></td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Del. ACKs</td>
<td></td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Socket</td>
<td></td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Concur. Conn.</td>
<td></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

(T1) = TCP-only, on x86 and AVR platforms  
(T2) = TCP-only, on ARM Cortex-M platform  
(T3) = TCP-only, on ARM Cortex-M0+ platform (NOTE: RAM usage for the same platform is ~2.5 kB for one TCP connection plus ~1.2 kB for each additional connection)  
(a) = includes IP, ICMP and TCP on x86 and AVR platforms  
(b) = the whole protocol stack on mbed  
(I) = interface inspired by POSIX  
Mult. = Multiple  
N/A = Not Available
Post-cutoff comments

• Yoshifumi Nishida (TCPM co-chair)
  – Comments:
    • Section 4.2.4. Cite draft-ietf-tcpm-rto-consider
    • Section 4.3.1. Need to clarify need of window size of 5 MSS to get 3 duplicate ACKs
    • Section 5.3. A typo
    • Section 5.3. TFO deviation from TCP semantics
    • Section 5.3. Discuss reducing TCP keep-alive interval
WGLC ?