TCB Control Block Sharing: 2140bis

draft-touch-tcpm-2140bis-00

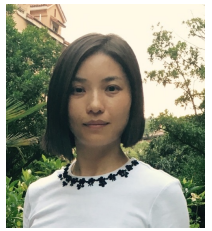
IETF 96 - Berlin

Joe Touch, USC/ISI

Michael Welzl, U. Oslo

Safiqul Islam, U. Oslo

Jianjie You, Huawei



Overview

- RFC2140 proposed TCP TCB sharing
  - Proposal at the time
- TCB sharing is now widely deployed
  - Useful to discuss experience, caveats
  - Useful to discuss relation to current protocols
- NB: replaces welzl-tcpm-tcb-sharing
  - That was a placeholder for changes

Changes from RFC2140

- Update to present tense
  - Cite more recent IW recommendations
  - Refer to current deployment
  - Add relation to later work: Cong. Manager (CM), MPTCP
- More detail
  - Add PMTU to list of cached values
  - Add equations in use for ssthresh sharing
- Clarification
  - Focus on parameter changes to existing and new state that result from connection start/end only (CM is “ongoing”)
  - Add caveats about impact and ECMP/LAG interaction

Deployment summary

TCB data	Status
old_MSS	Cached and shared in FreeBSD
old_RTT	Cached and shared in FreeBSD
old_RTTvar	Cached and shared in FreeBSD
old_snd_cwnd	Not shared
old_ssthresh	Cached and shared in FreeBSD and Linux: FreeBSD: arithmetic mean of ssthresh and previous value if a previous value exists; Linux: depending on state, max(cwnd/2, ssthresh) in most cases

Caveats summary

- Impact
  - Directly improves only “goldilocks” connections
    - Not too short, not too long
  - Indirectly improves endpoint and network efficiency
    - Reduces fighting “on the wire” to provide initial feedback
- Endpoint pair issues
  - ECMP, LAG may reduce utility for endpoint pair
  - When known to share bottleneck (e.g. VPN – encapsulation, ESP encryption), share within SYN dest port of a pair

Issues for WG

- Path for adoption
  - Previously individual, informational
  - This update seems appropriate for BCP
- Request for WG adoption
  - TCPM of course ;-)
- WG task
  - Decide specific MAY/SHOULDs and negatives
  - MUSTs are unlikely – this should stay optional