Careful resumption of congestion control from retained state with QUIC

draft-kuhn-quic-careful-resume-02

N. Kuhn, E. Stephan, G. Fairhurst, T. Jones, C. Huitema
Catching-up with the activity

• Need for quick ramp-up for high BDP paths

Storing previous path characteristics (rate, rtt, etc).
Deciding to use information to initialise a new connection
Being careful to « undo » if there any problem is found
Catching up with the activity

- QUIC WG adoption call of
  - draft-kuhn-quic-careful-resume-02
    - Store computed congestion control parameters of a given connection
    - Modify the congestion control behavior of a subsequent connection
  - draft-kuhn-quic-bdpframe-extension-00
    - BDP Frame extension for QUIC
    - How to store and exchange computed congestion control parameters

- Current design uses QUIC (see draft-kuhn-quic-bdpframe-extension)
  - “Problem space [...] more broadly to do with congestion control and not entirely QUIC specific”
  - The final CC solution must effectively share capacity with all transports
  - It could be possible to find a way to do this with other transports?
CC Discussion items – relevant for TSVWG

• Assume previous connection finished and stored information.
• What relevant information can be got to check the path?
  • packet loss after sending the IW
  • minRTT
  • inter-packet time
• How does a server “carefully” jump to utilise previously estimated capacity?
• How does it detect any congestion and react after the jump?
• What aspects are congestion-control specific?
  • Eg. CUBIC vs Reno
  • E.g. impact of HYSTART / HYSTART ++

NB: Currently discussed in draft-kuhn-quic-careful-resume-02
Other topics
- related to specific transports

• *How does client ask* for this jump?
  • QUIC 0-RTT and TLS certificate signaling
  • Congestion control and TLS dependent
  • For QUIC,
    • draft-kuhn-quic-careful-resume-02 compares three approaches
    • draft-kuhn-quic-bdpframe-extension-00 details design of one approach
  • What do I send and when do I send it?

• Real system performance
Summary

- Is there appetite for the CC part of this work in TSVWG?
- Who would join us as we seek to define appropriate details?
Catching-up with the activity (QUIC)

• BDP FRAME extension for QUIC

1. During a previous session, current RTT (current_rtt), CWND (current_cwnd) and client’s current IP (current_client_ip) are stored as saved_rtt, saved_cwnd and saved_client_ip;

2. When resuming a session, the server might set the current_rtt and the current_cwnd to the saved_rtt and saved_cwnd of a previous connection.


Implemented in PICOQUIC by F. Simo, D. Pradas
https://github.com/private-octopus/picoquic/pull/1209