

Pacing in Transport Protocols

draft-welzl-iccrp-pacing-01

Michael Welzl, Wesley Eddy, Vidhi Goel, Michael Tüxen

ICCRG
IETF 121

Context – what is this about?

- Give guidance to implementers via:
 - Discussion of general considerations and consequences
 - An overview of how others do it
- -00 draft already described Linux TCP and QUIC BBR
 - Linux TCP based on: <https://tinyurl.com/26698df9>
 - QUIC BBR based on open source implementations: Google's *quiche* and Meta's *mvfst*

News

- -01 version has two new authors: Vidhi Goel and Michael Tüxen
 - Who contributed text about
 - Apple OSes
 - FreeBSD
- ... both to be extended / refined in future versions.

Apple OSes

- Upper layer protocols / apps. can pace in a way that suits them
 - For example, a stream based protocol like TCP might pace packets differently than a video conferencing app.
- Private API since iOS 17 and macOS 14
- Application / transport protocol computes & sets timestamp for each packet
- AQM will delay packets accordingly, with 3 second upper limit

FreeBSD

- Multiple TCP stacks
 - RACK and BBR stacks support pacing using the TCP High Precision Timer System (HPTS) (kernel loadable module)
- Generally, `tcp_output()` is called when:
 - TCP segment is received,
 - the application provides new data to send, or
 - a timer fires
- When pacing (RACK or BBR stack) is used, `tcp_output()` can also be called by HPTS:
 - `tcp_output()` sends a micro burst and schedules itself for being called after the interburst send time using the HPTS
- RACK stack: app can set pacing rate & max. burst size with TCP socket options
 - Stack then uses these values to compute the actual micro burst size and inter-burst time
 - HPTS can handle many connections, and `tcp_output()` optimized for being called more often than the `tcp_output()` function of the default stack. This allows TSO usage.

Main next steps (open issues in github)

- More detail on Apple OSes and FreeBSD
- Discussion of initial RTT estimate in 5G (thanks Ingemar Johannson)
- Discussion of interactions between TCP and application pacing (an upper limit, as with video streaming)

Thank you!

Your feedback is very welcome:

<https://github.com/mwelzl/draft-iccrg-pacing>

Questions?