

# Transport parameters for QUIC 0-RTT connections

Nicolas Kuhn, Emile Stephan, Gorrry Fairhurst, Tom Jones

QUIC WG, IETF-109

# The Proposal

- The idea in a nutshell
  - Remembering and exchanging additional parameters when reconnecting with 0-RTT
    - BW estimation based on `inflight_data`
    - RTT estimation based on `min_rtt`
- Use-case
  - Sharing server's estimation of path parameters so that clients can adapt their requests
  - Improving ramp up with 0-RTT on the server
- Several implementations
  - Ours using PICOQUIC with TLS1.3
  - Matt Joras' using "BDP\_TOKEN"
- Short term objective:
  - Merge the proposed ideas in single revised draft

# Expected Benefits

- Evaluations based on
  - draft-kuhn-quic-4-sat-06 scenarios
  - Implementation of draft-kuhn-quic-0rtt-bdp-07
  - Picoquic : <https://github.com/private-octopus/picoquic/pull/1073>
- Network characteristics:
  - 50 Mbps download / 10 Mbps upload
  - RTT : 650 ms
- Congestion Control (CC)
  - CUBIC
  - 0-RTT-BDP reaction:
    - Jump to a preciously measured capacity
    - (not recommended, but “easy to implement” as a first step)
    - Beware of the potential issue in using bytes\_in\_flight metric
- Application level
  - 2 MB transfer - median

Without 0-RTT	With 0-RTT	With 0-RTT-BDP
4,3 s	3,4 s	2,9 s

# Open Discussion: Exchanging parameters

- BDP\_TOKEN vs NEW\_TOKEN
  - Any views ?

# Open Discussion: CC Reaction

- Avoid CC overshoots
  - Same security as max\_initial\_data
  - ““I received at 1 Gbps last time”, when in fact it can only absorb 10 Mbps”
    - Need to add mechanism to protect the integrity when client pushes a token back
- Algorithm proposed in 0-rtt-bdp draft
  - If reception of IW is confirmed for the first RTT of data (no loss)
  - If path is “similar” to a recent previous session (e.g., similar RTT)
  - Then, a sender can use the previous path information as an input to help determine a new safe rate (e.g., with pacing)
- Questions ? Comments ?

# Next Steps

- Discuss and agree on a method
  - Revise draft
  - Test
  - More feedback
- Revise draft to describe the solution
  - Add in the interop matrix