

# Update on draft-ietf-netconf-over-quic

Jinyou Dai, Shaohua Yu, Weiqiang Cheng, Marc Blanchet, Per Andersson

IETF 121 Dublin, November 2024

# Updates

- draft-dai-netconf-quic-netconf-over-quic was adopted by the NETCONF WG on September 20th, 2024
- draft-dai-netconf-quic-netconf-over-quic-06 == draft-ietf-netconf-over-quic-00
  - As is, no changes
- The draft is under the netconf wg GitHub repo. Please submit issues or PR
  - <https://github.com/netconf-wg/netconf-over-quic>.
- During the adoption call and after, the following comments were processed and folded into draft-ietf-netconf-over-quic-01

# Changes Made into -01

- See Git log
- Editorial or mostly editorial
  - Abstract tightened up
  - Used « server » and « client » instead of « manager » and « agent »
  - Rephrased so that idle timeout to be optionally set
  - Replaced NETCONFoQUIC with NETCONF over QUIC/noq
  - Removed redundant/basic text about QUIC/NETCONF
  - Replaced NoQ to 'noq' tag to align with RFC9250 for IANA Application Layer Protocol Negotiation (ALPN) Protocol IDs registry
  - Removed historic text (reference to historic transports)
  - Applied various wordsmiting fixes (capitalize keywords, spaces, authors meta data, ...)

# Remaining Comments

- Issue #7: avoid repeating parts of specs that are already in the base NETCONF and QUIC
  - We did a lot. Kept some to give some context about QUIC ways to deal with streams which is used for NETCONF. Please comment if more cuts are needed
- Issue #6: leverage RFC 9250
  - Did not have time to look to see what to leverage. Would be useful if more details of the intent would be provided.
- “Just in” (did not have time to discuss, but seems reasonable)
  - Issue #18: Clarify ambiguous QUIC stream mapping
  - Issue #19: Clarify authentication

# Implementation

- Proxy POC implementation underway (by Adolfo Ochagavia) with the following setup:
  - NETCONF client - ssh - proxy - quic - proxy - ssh - NETCONF server
  - Simpler as the proxy has limited need to know the NETCONF protocol
  - Base flow working. Some comments on the spec (see #18, #19).
  - Details:
    - <hello> exchanged in a bidirectional stream, started by the client. Framing is implicit (provided by the stream abstraction)
    - <rpc> and <rpc-reply> exchanged in bidirectional streams, started by the client (each request gets its own stream). Framing is implicit.
    - <notification> events are received in a single unidirectional stream, started by the server upon subscription.
  - Should each notification be in its own unidirectional stream (if yes, then notifications are not guaranteed to arrive in order)?
  - Should a single stream be used by all rpc calls and replies?
  - (Context: streams are essentially free in QUIC)
  - More testing coming

# Next Steps

- Short draft.
- Intend to fix remaining issues before next IETF. Please comment
- More feedback from implementations