

# Multipath extension for QUIC

## Draft-ietf-quic-multipath-15

QUIC meeting @ IETF-123 Madrid

Yanmei Liu, Yunfei Ma, Quentin De Coninck,  
Olivier Bonaventure, Christian Huitema, Mirja Kühlewind

# Agenda

- ❖ Brief Summary: Changes from draft -14 to draft -15
  - -14 was a big editorial clean-up before WGLC
- ❖ Remaining Issues
- ❖ Final Steps

# Diff from -14 to -15

- Renamed status frames and transport error codes used in the PATH\_ABANDON frame (next slide)
- Fixed examples of new path establishment and AEAD nonce calculation
- Various Clarifications (which imply some changes in normative language):
  - It's not recommend to send keep-alives as it can create unnecessary overhead, especially if there are other actively used paths.
  - Connection closure follows RFC 9000 but which PTO? Closing and draining connection states "SHOULD persist for at least three times the largest PTO among all paths"
  - "If an endpoint sends a PATH\_RESPONSE, it MUST be sent on the same path as used by the packet that contained the PATH\_CHALLENGE frame, using a connection ID associated with the same path ID"
  - Migration (4-tuple change within path ID) "MUST be treated as path migration, as specified in Section 9.3 of [QUIC-TRANSPORT], with the constraint that all connection IDs used during path migration MUST be associated with the current path ID of the path being migrated"
  - "with the multipath extension multiple paths could use the same 4-tuple but might have different MPS"
- Improved discussion of DoS attacks
- And a bunch of editorial improvements (Thanks to all contributors!)

# Renaming Status Frames and Error Codes

## Frames

- PATH\_AVAILABLE -> PATH\_STATUS\_AVAILABLE
- PATH\_BACKUP -> PATH\_STATUS\_BACKUP

## Error codes

- APPLICATION\_ABANDON -> APPLICATION\_ABANDON\_PATH
- RESOURCE\_LIMIT\_REACHED -> PATH\_RESOURCE\_LIMIT\_REACHED
- UNSTABLE\_INTERFACE -> PATH\_UNSTABLE\_INTERFACE
- NO\_CID\_AVAILABLE -> NO\_CIS\_AVAILABLE\_FOR\_PATH

# Issue [#557](#): Missing PATH\_ABANDON codes

Should PATH\_UNSTABLE\_INTERFACE be split up in more error codes? E.g.

- COMMUNICATION\_FAILED and POOR\_PERFORMANCE
  - These cases are currently bit covered by PATH\_UNSTABLE\_INTERFACE". Is there a real difference?
- TIMED\_OUT
  - For the condition "path as not been used for a long time, don't need it anymore." APPLICATION\_ABANDON could/should be used as this is a regular close and not an error.

Proposed action: none (except maybe renaming the current error code?)

## Issue [#558](#): Uncoordinated abandonment of paths

Do we need provide more guidance on the “uncoordinated abandonment” case for two paths on the same 4-tuple? Or just remove this guidance at all?

Currently the text says:

Uncoordinated abandon of both endpoints may result in deleting both paths instead of just one. To avoid this pitfall, endpoints could adopt a simple coordination rule, such as only letting the client initiate closure of duplicate paths, or perhaps relying on the application protocol to decide which paths should be closed.

# Issue [#550](#): Why isn't NEW\_TOKEN path-specific?

## Current text:

As specified in [Section 9.3](#) of [\[QUIC-TRANSPORT\]](#), the server is expected to send a new address validation token to a client following the successful validation of a new client address. The client will receive several tokens. When considering using a token for subsequent connections, it might be difficult for the client to pick the "right" token among multiple tokens obtained in a previous connection. The client is likely to fall back to the strategy specified in [Section 8.1.3](#) of [\[QUIC-TRANSPORT\]](#), i.e., pick the last received token. To avoid issues when clients make the "wrong" choice, a server SHOULD issue tokens that are capable of validating any of the previously validated addresses. Further guidance on token usage can be found in [Section 8.1.3](#) of [\[QUIC-TRANSPORT\]](#).

**Question:** Is it acceptable to have the token recipient assume that the token vendor is able to remember EVERY address that was in use?

**Counter proposal:** Have token bound to path

This was discussed ([#248](#)): Do we need to change? Or just document assumptions better?

## Two editorial issues

- [#586](#): Migration/Probing (was: Use of "data packet")
- [#556](#): 3 PTO requirement for PATH\_ABANDON

# Final steps

- WGLC passed
  - Thanks to all reviewers, especially Martin Thomson and Tommy Pauly!
- Latest revision addresses most of the issues
- New revision planned after the meeting to address remaining 3-5 issues