

Multipath extension for QUIC

draft-ietf-quic-multipath-06

QUIC meeting @ IETF-118 Prague
Yanmei Liu, Yunfei Ma, Quentin De Coninck,
Olivier Bonaventure, Christian Huitema, Mirja Kühlewind

Agenda

- ❖ Diff from -05 to -06
- ❖ Interop reports for 06
- ❖ Open issues
- ❖ Next steps

Diff from -05 to -06 (Most important)

1. New Frame Types: PATH_STANDBY and PATH_AVAILABLE

- Split PATH_STATUS frame into 2 new frames: PATH_STANDBY and PATH_AVAILABLE frames share a common sequence number space (as discussed at 117)
- No change in semantic: "standby" and "available" status of the PATH_STATUS are the same states as provided by PATH_STANDBY and PATH_AVAILABLE
- PATH_STANDBY and PATH_AVAILABLE can be sent for any issued CID (even if not in active issue) -> potential clarification needed (issue [#283](#))
- Path states are recommendations from the receiver to sender about path usage for sending data
 - The recommendation might be different in each direction
 - It is not binding, and local preferences might override
 - You might anyway send control frames incl. ACKs, e.g. if the other end is actively using the path for sending data
 - This points need clarification!

Diff from -05 to -06

2. Any frame can be sent on a new path at any time when the anti-amplification limits and the congestion control limits are respected.

- Which means endpoints are allowed to send path available/standby frames before the path validation complete
- Also the probing frames, the stream / datagram frames...

Share the same security consideration with QUIC-Transport

Diff from -05 to -06

3. Expand scenes in which client uses only one IP address and server listens on only one IP address either.

- As a path is determined by 4-tuple, endpoints could use different ports with the same IP address when using multi-path extension.

Interop reports for -06

More features have been tested compared to last time:

- Path Status (S)
- Key Update (U)

More support of Path Close (C)

A number of bugs was found in the various implementation and one spec issue filed:

<https://github.com/quicwg/multipath/issues/290>

Active Participation by implementers of PICOQUIC, XQUIC, RASK and QUICHE

<i>server</i>				
<i>client</i> ↓	xquic	picoquic	Rask	quiche
xquic	HVDCSUA	HVDSUA	HVDCUA	HVDCSUA
picoquic	HVDUA	HVDIUA	HVDUA	HVDIUA
Rask	HVDCSU	HVDCSU	HVDCSU	HVDCSU
quiche	HVDCSUA	HVDACIS	HVDA	HVDACIS

Feature	code	details
Handshake	H	The handshake completes with successful negotiation
Path Validation	V	Client sends PATH_CHALLENGE
Send data	D	Stream data is sent on all paths;
Path Close	C	Client closes a path with PATH_ABANDON frame

Feature	code	details
CID change	I	Upon some events, the client uses the new server CID
Path status	S	Client sends PATH_AVAILABLE and PATH_STANDBY
Key Update	U	One endpoint updates keys
Multipath ACK	A	Sends ACK (randomly) on all path
CID retirement	R	One endpoint send an RETIRE_CONNECTION_ID

Issues: [#179](#) / [#214](#) on Path ID

Key issue: In the current draft, each CID is linked to a different packet number space. If **CID rotation** on a path happens, it affects the following mechanisms:

- Packet encryption / decryption: sequence number of CID is part of the nonce
- Loss detection and retransmission: if a packet threshold is used for loss detection, the new and old packet number space (before / after CID rotation) may need to be considered
- If a NAT rebinding happens at the same time as CID rotation, this is considered a new path (and the old path stays open but is not working anymore) - see also issue [#188](#)

While CID rotation is assumed to be a rare, avoiding CID rotation entirely is not an option to avoid linkability for migration inline with RFC9000 (see closed issue [#273](#)).

Alternative proposal: separate Path IDs from Connection IDs: [#214](#) (see also [#179](#))

- Introduce an explicit path ID that stays constant even if the CID on a “path” changes
- Needs new frames for CID management (NEW_CONNECTION_ID, RETIRE_CONNECTION_ID) and more per-path state

Question: How important/likely are these cases vs additional state/frames?

Next Steps

- Plan for becoming Proposed Standard
- Solve the left issues
- More implementation and interop tests
 - Hackathon from 117 to 119
 - Have a [Pull Request](#) about adding multipath cases in the Interop Runner
- WG Last Call at IETF 119?