Multipath extension for QUIC

Draft-ietf-quic-multipath-11

QUIC meeting @ IETF-121 Dublin

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

Agenda

- Brief Summary: Changes from draft -10 to draft -11
- Hackathon Interop Reports: draft -11
- Open issues
- Next Step

Diff from -10 to -11(Semantics change)

1. Allow endpoints to send packets without challenge if the 4-tuple address has already been validated
2. Re-define the Error Code field of PATH_ABANDON frame, and delete the Reason Phrase field
3. Add PATHS_BLOCKED frame to tell the peer that the maximum path identifier limit is reached

Update transport parameter to 0x0f739bbc1b666d11 (draft-11).

https://datatracker.ietf.org/doc/html/draft-ietf-quic-multipath-11

Diff from -10 to -11 (Frame Updates)

Prev Frame Name (draft-10)	**New Frame Name (draft-11)**	**Usage**
MP_ACK	PATH_ACK	Acknowledging packets on multiple paths.
MP_NEW_CONNECTION_ID	PATH_NEW_CONNECTION_ID	Endpoints issuing new Connection IDs for Paths.
MP_RETIRE_CONNECTION_ID	PATH_RETIRE_CONNECTION_ID	Used by endpoints to retire Connection IDs for Paths.
PATH_STANDBY	PATH_BACKUP	Used by endpoints to inform the peer about its preference to not use the indicated path for sending.
N/A	PATHS_BLOCKED `(type=0x15228c0d)`	Endpoints SHOULD send a PATHS_BLOCKED frame when it wishes to open a path, but it's limited by peer's maximum path identifier limit.

Diff from -10 to -11 (Editorial)

1. Add more guidance on using multiple paths with the same 4-tuple
2. Claim that Connection ID sequence number space starts at 0 for all paths
3. Add more implementation comments for key rotation, more guidance on when to use PATH_ACK frames

Hackathon Interop Reports: draft-11

server / client ↓	xquic	picoquic	Rask	quiche	mp-aioquic
xquic	HVDCISURAB	HVDCIURAB	HVDCISURAB	HVDCISURA	HVDCISURB
picoquic					
Rask	HVDCSURMB	HVDCSUR	HVDCSURMB	HVDCSRM	HVDCSURB
quiche	HVDISRA	HVDISRA	HVDCISRA	HVDCISARM	HVDISRA
mp-aioquic	HVDISR	HVDISR	HVDISR	HVDISR	HVDCSR

Core Features Tested

Feature	code	details
Handshake	H	The handshake completes with successful negotiation of enable_multipath transport parameter (both ends indicate 0x01)
Path Validation	V	Client sends PATH_CHALLENGE frame to open a new path and server replies with PATH_RESPONSE
Send data	D	Stream data (of one of more streams) is send on all paths; ACK_MP frames are sent and processed
Path Close	C	Client closes a path with PATH_ABANDON frame

Optional Features Tested

Feature	code	details
CID change	I	A server offers new CIDs for a path using PATH_NEW_CONNECTION_ID to a client in advance. Upon some events, the client starts using a new server CID on one path
Path status	S	Client sends PATH_BACKUP and PATH_AVAILABLE frames
Key Update	U	One endpoint updates keys and sends at least one packet with the new key on all active paths
Multipath ACK	A	One endpoint sends data and the other endpoints sends PATH_ACK (randomly) on all path independent of where data is received
CID retirement	R	One endpoint send an PATH_RETIRE_CONNECTION_ID for an active path
Migration	M	Change CID and 4-tuple e.g. port on an existing path
Paths Blocked	B	Server configures a path limitation of two paths, client is configured to open three or more paths. When unable to open requested number of paths it sends PATHS_BLOCKED and server logs or increases number of paths by announcing more using PATH_NEW_CONNECTION_IDS.

Issues with PRs

- Issue #455 : The last sentence in the “PATH_ABANDON frame” section says: *After sending the PATH_ABANDON frame, the endpoint MUST NOT send frames that use the Path ID anymore, even on other network paths.* This is not correct, contradicts the existing text in the “Path Close” section.
  - PR #456: Replace this incorrect sentence by a reference to the “Path Close” section.

- Path abandon Error code:
  - PR #462: Change the experimental values for 2 error types, fix into 62-bit variable integer

- Issue #461: PATH_BLOCKED is informational
  - PR #463: Clarification that the PATH_BLOCKED frame doesn’t mandate a specific action but it for information. Endpoints send MAX_PATH_ID frame when they decide there’s not enough unused Path IDs for new paths.

Issue #459: What happen if endpoint doesn't issue CIDs for each available Path ID?

- Current draft: In order to let the peer open new paths, **it is RECOMMENDED to proactively issue a Connection ID for at least one unused Path ID.**

- Why an endpoint might delay provision of CIDs:
  - The PATH_NEW_CID frames can only be sent after the MAX_PATH_ID is negotiated, so there will always be a delay (longer delay if lost…)

- Do we want to redesign the PATH_BLOCKED frame to add field for this issue? (e.g no available cid field for specified Path ID)

Issue #414: Error code for Path Abandon (PR #445)

- Error Code for Path Closure:

```
PATH_ABANDON Frame {
  Type (i) = TBD-02 (experiments use 0x15228c05),
  Path Identifier (i),
  Error Code (i),
}
```

Code	Description	Specification
APPLICATION_ABANDON	Path abandoned at the application's request	Section 8
RESOURCE_LIMIT_REACHED	Path abandoned due to resource limitations in the transport	Section 8

- PR #462: Fix experimental values for error code field, within 62-bit variable integer
- Do we need more codes(e.g. UNSTABLE_INTERFACE / NO_CID_AVAILABLE)?

Issue #457: Handling of PTO

- Scenario: Endpoint sent packets on one path, suddenly the path went into black hole. Following RFC 9002, as the endpoint can't get any acknowledges back from this path, these packets are not considered lost until a PTO occurs.

- One potential benefit for multipath: Endpoint could choose to retransmit the frames in these packets on another available path, but it depends on these packets considered lost first. When to trigger?

Issue #458: Path migration or creating a new path?

- Scenario: Should we support Path Migration after multipath is negotiated?
  - Endpoint wants to use the new 4-tuple address for traffic transmission
  - **It's RECOMMENDED to create a new path with a new Path ID for supporting the use case**
- When endpoint receive a new packet, it decides based on the packet DCID:
  - Same path ID, same 4-tuple, using the same path
  - Same path ID, different 4-tuple, always treated as NAT Rebinding
  - Same path ID, different CID, always treated as CID rotation
- Implementation Guidance for {Same path ID, different 4-tuple}
  - NAT rebinding for default, which means stop sending on the old address immediately after the new address is validated.
  - (Choice A) Choose to create new path to avoid the complexity of switching path context
  - (Choice B) Accept the complexity and keep compatible with path migration
- Related issue #460: Using prefered address is considered as path migration event in draft-11

Next Step

- WG last call?