Multipath extension for QUIC
Draft-ietf-quic-multipath
Explicit Path ID Proposal

QUIC meeting @ IETF-119 Brisbane
Yanmei Liu, Yunfei Ma, Quentin De Coninck,
Olivier Bonaventure, Christian Huiitema, Mirja Kühlewind
Agenda

- Background: Key issue that Explicit Path ID want to solve
- How Explicit Path ID works
- Pros and Cons
- Interop reports
- Open issues
Key issue (ietf 118)

Problem: The implicit approach (-06) is using an Identifier which doesn’t have the same life time as the network path

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
How Explicit Path ID works (PR #292)

- Explicit Path Identifier used to identify a path in the connection; idea proposed by Marten (issue #214)

<table>
<thead>
<tr>
<th>Path Management</th>
<th>Explicit Path Identifier to identify a path in a connection</th>
<th>CID Sequence Number as Path Identifier</th>
</tr>
</thead>
</table>
| CID Management (Control Frames) | - MP_NEW_CID frame ties CID to Path ID and CID Sequence number per Path-ID  
- MP_RETIRE_CONNECTION_ID frame specifies both Path ID and Sequence number | RFC9000 CID management |
| Packet Number Space | Packet Number Space is bound to Path ID and remains stable when CID rotation happens | Packet Number Space changes with CID rotation |
## Pros and Cons

<table>
<thead>
<tr>
<th></th>
<th>Explicit Path ID (PR#292)</th>
<th>Draft-06</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Path management</strong></td>
<td><strong>Pro:</strong> Link between incoming packet and path is unambiguous</td>
<td><strong>Con:</strong> Need to treat situations when CID rotation / NAT rebinding happens</td>
</tr>
<tr>
<td><strong>CID management</strong></td>
<td><strong>Con:</strong> Increases complexity to:</td>
<td><strong>Pro:</strong> Same as RFC 9000</td>
</tr>
<tr>
<td></td>
<td>● maintain CIDs per path</td>
<td></td>
</tr>
<tr>
<td></td>
<td>● manage CID/Path-ID lifetime</td>
<td></td>
</tr>
<tr>
<td><strong>PN state: loss recovery and congestion control</strong></td>
<td><strong>Pro:</strong> Loss recovery and congestion control can rely on single sequence number space for the duration of the path</td>
<td><strong>Con:</strong> CID Renewal triggers use of a new number space which makes loss recovery potentially more complex</td>
</tr>
</tbody>
</table>
Explicit Path ID management works well
Clear logic reduces code
Do we want to merge PR #292 (Explicit Path ID)?
Open Issue that have a proposed solution

#297. Path ID should not be reused.
- Path ID is generated monotonically increasing. It’s limited by MAX_PATHS.
- Once a path is abandoned. The Path ID MUST NOT be reused in any other paths.

PR: PR #315

#317. Should server preferred address have its own path ID?
- Yes, use Path ID 1

#294. "Path ID" needs to be clarified. Do both endpoints use the same path ID, or independently choose which path ID to use?
- Yes, use the same Path ID for both sides
- Two options to coordinate use of numbers:
  - Only allow the client to initiate paths
  - Divide path ID space between client and server -> see next slides
Issue #47: Should servers be allowed to open new paths?

If we want to support server-oriented paths with explicit Path ID:

- Need to use even / odd Path IDs to distinguish between client-initiated / server-initiated (like bidi streams)
- **Transport Parameters: Initial_max_paths**
  - Client sends `Initial_max_paths` to indicate the initial max odd Path ID which is allowed to initialize by the server side
  - Server sends `Initial_max_paths` to indicate the initial max even Path ID which is allowed to initialize by the client side
- **MAX_PATHS frames**
  - Client sends `MAX_PATHS` frame to inform the max odd Path ID which is allowed to initialize by the server side
  - Server sends `MAX_PATHS` frame to inform the max even Path ID which is allowed to initialize by the client side
  - Need to add a type field("client-initiated / server-initiated") for `MAX_PATHS` frames
Open Issues: How do we retire CID of all paths?

Issue: [#295] / [#313]

How to retire a Path ID?

- Endpoint sends PATH_ABANDON frame to request the peer to stop sending packets with the specific Path ID
- The peer SHOULD also send PATH_ABANDON frame for that Path ID once it received the PATH_ABANDON frame

How do we retire all CIDs of the corresponding path?

- PATH_ABANDON also triggers the CID retirement of all the CIDs allocated for the corresponding Path ID
- Endpoints SHOULD send MP_RETIRE_CONN_ID after 3 PTOs
AEAD and Hardware offloading

AEAD Decryption / Encryption
- The nonce of AEAD is calculated by combining the packet protection IV with the packet number and with the least significant 32 bits of the path identifier pre-allocated for the Destination Connection ID.

Hardware Offloads ([Issue #25](#))
- In order to not change the hardware and still support multipath, the QUIC kernel module and/or driver must XOR in the destination connID sequence number to the IV.

- **Explicit Path ID or Sequence of DCID?**
  From a hardware perspective it doesn't matter. The nonce construction would be the same as I detailed above with the driver XOR'ing in the connid_seq_num or the path_id into the IV before offloading the flow. (comments by Eric)

---

**Example in Multi-path Draft**

IV: 0x6b26114b9c63a9e8dd4f
Connection ID Sequence Number: 0x3
Packet Number = 0xaead
New IV passed in flow offload = (IV XOR (connid_seq_num << 64))
0x6b26114b9c63a9e8dd4f (IV)
XOR 0x000000030000000000000000 (connid_seq_num << 64)
--------------------------
0x6b2611489c63a9e8dd4f
Nonce (hardware): (IV XOR pkt_num)
0x6b2611489c63a9e8dd4f (offloaded IV)
XOR 0x000000000000000000000000aead (pkt_num)
--------------------------
0x6b2611489c63a9e873e2