Multipath extension for QUIC

draft-ietf-quic-multipath-01

QUIC meeting @ IETF-113 Vienna

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

Status

- draft-ietf-quic-multipath-00: submitted in Jan 2022
  - Update on negotiation table (already presented at last meeting)
  - Clarifications on packet number spaces and ACK delays
  - Added definition of the term path
- draft-ietf-quic-multipath-01: submitted in March 2022
  - High-level overview about multipath extension added
  - Clarify that transport parameters from RFC9000 remain unchanged
  - Clarification on idle timeout for paths
  - Some recovery and ACK Delay Considerations for multiple paths
  - Implementation considerations to handle different PMTU sizes
  - And some editorial modifications such as updating path closure figure and descriptions, etc.

Open design issues

1. Are we sure we really don't need stand-by? #22
2. Should servers be allowed to open new paths? #47
3. Sending non-probe packets before path validation complete #50
4. ECN support and single/multiple packet number spaces #87
   - PR New text on ECN handling with single PNS #97
5. Do we need a transport parameter to negotiate max path idle timeout? #95
6. Choosing between a single packet number space vs. multiple packet number spaces #96
   - PR First shot at a unified proposal #103
   - Subissue: Multiple packet number spaces can be compatible with zero-length Connection IDs #27
     - PR Update identification of packet number space identifier #29

Are we sure we really don't need stand-by? #22

- One peer may want to **indicate** to the other peer that a path should only be used as **“back-up”**
- Typical handover case:
  - Client opens both a wifi and a cellular path
  - Client sends keep-alive on the cellular path to make sure one can switch over quickly
  - Client requests data from the server
  - But server doesn't know which path to use (or that it should not use one of paths as it is more costly)
- This is related to packet scheduling but might still be in scope for this base extension draft as it needs some **signaling**
  - draft-liu had the PATH_STATUS frame to indicate the "standby" or "available"
  - (See also B flag in MPTCP MP_PRIO option)
- Do we want to (re-)add something or is that a separate extension?

Should servers be allowed to open new paths? #47

- In RFC9000 path migration is restricted to clients only (mainly because of problems with NATs).
- However, with multipath, failure of opening a new path is less critical as the old path(s) is not abandoned at the same time.

- Can we release this restriction in the multipath extension?
  - Or is there no good reasons to keep it?
  - This would support additional use cases.

Sending non-probe packets before path validation complete #50

- Path validation brings at least one round-trip-time delay for data to be sent on the new path

- Can we add a mechanism similar to 0-RTT transmission that still avoids amplification attacks?

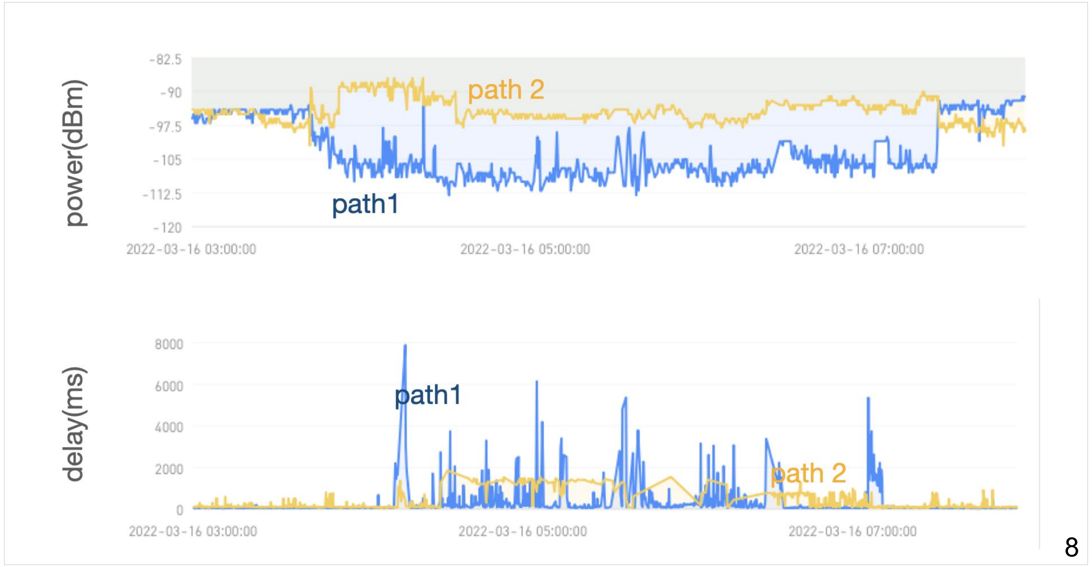
ECN support and single/multiple packet number spaces #87

- With a **single packet number space**, the ECN fields of the ACK frame provide ECN information that is common to all paths.
- This makes it impossible for a sender to determine which path is affected by congestion when it receives an ACK frame with positive CE counter

- PR New text on ECN handling with single PNS #97
  - Recommendation: `Hosts` **`SHOULD only acknowledge packets in the same ACK frame from one path`** `if an ECN counter increased (especially the CE counter)`
  - Conservative reaction: `If a host receives an ACK frame that increases the CE counter and contains packets from different paths, it` **`MUST treat the CE marking as if it was received on either of the path.`**
  - Also: `An host that receives an ACK with an ECN counter increase acknowledging packets from different paths` **`MAY disable ECN marking`** `and send all subsequent packets as Not-ECN capable.`
- Alternatives: Don’t use ECN, or only on one path.

Do we need a transport parameter to negotiate max path idle timeout? #95

- Currently `max_idle_timeout` is also used for each path to close on idle time-out.
- Do we need to signal separate time-out values per path?
- Discussion so far:
  - Yes, "path idle timeout" might be a way to have a stronger guarantee to stop using paths (closing them)
  - No, just makes the protocol more complex; just use a shorter time-out locally (if it’s only one of multiple paths)



Choosing between a single packet number space vs. multiple packet number spaces #96

Quick recap:

	Single PN space	Multiple PN spaces
Efficiency	Almost similar efficient if loss detection is adapted to use "order of sending per path"	More efficient, due to complete reuse of loss-recovery logic and no additional state
Code Complexity	No new code path if some inefficiency is acceptable. Requires substantial additional code to manage the ACK size and loss recovery efficiently	Multiple instantiations of the loss recovery algorithm for each path
ACK handling	Without special logic, ACKs can be much larger. Additional logic needed needs to be standardized.	New ACK Frame keeps small-sized ACK for each path; ACK-Delay and ACK-ECN work as expected with changes needed
Zero-length CID	Supported	Not supported

Choosing between a single packet number space vs. multiple packet number spaces #96

So we are left with the "zero-length CID" issue in the multiple PN space case:

Client SCID	Server SCID	Support	Priority/Use cases
long	long	Supported in **both variants**	Used by many implementations
NULL	long	Requires **special support in multiple spaces case**; could work but might be fragile	Preferred configuration of many big deployments
long	NULL	Requires **special support in multiple spaces case**; could work but might be fragile	Rarely used, server load balancing does not work
NULL	NULL	**Does not work for multiple spaces**	Only mentioned in some P2P deployments

Choosing between a single packet number space vs. multiple packet number spaces #96

New PR: First shot at a unified proposal #103

- Mandatory support of multiple packet number spaces (and ACK_MP frame), if multipath extension is negotiated
- Support for zero-length CID (at sender-side) is optional
    - The receiver of packet without CID (zero-length) sets the number space ID to 0 in ACK_MP
    - Senders that use multiple paths and send packets without CID have to implement additional logic to minimize the impact of multipath delivery on loss detection and congestion control/ECN handling
        - Or alternatively sender only uses one path at a time…

Client SCID	Server SCID	What
long	long	Multiple number spaces
NULL	long	Multiple number spaces on client side (one per CID), single space on server side
long	NULL	Multiple number spaces on server side (one per CID), single space on client side
NULL	NULL	Single number space on each side