Workgroup: QUIC Working Group
Internet-Draft: draft-ietf-quic-multipath-07
Published: 10 April 2024
Intended Status: Standards Track
Expires: 12 October 2024
Authors: (Y. Liu), Ed. (Y. Ma)
Alibaba Inc. Uber Technologies Inc.
Q. De Coninck, Ed. O. Bonaventure
University of Mons (UMONS) UCLouvain and Tessares
C. Huitema M. Kuehlewind, Ed.
Private Octopus Inc. Ericsson

Multipath Extension for QUIC

Abstract

This document specifies a multipath extension for the QUIC protocol to enable the simultaneous usage of multiple paths for a single connection.

Discussion Venues

This note is to be removed before publishing as an RFC.

Discussion of this document takes place on the QUIC Working Group mailing list (quic@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/quic/.

Source for this draft and an issue tracker can be found at https://github.com/mirjak/draft-lmbdhk-quic-multipath.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 12 October 2024.

Table of Contents

1. Introduction

This document specifies an extension to QUIC version 1 [QUIC-TRANSPORT] to enable the simultaneous usage of multiple paths for a single connection.

This proposal is based on several basic design points:

- Re-use as much as possible mechanisms of QUIC version 1. In particular, this proposal uses path validation as specified for QUIC version 1 and aims to re-use as much as possible of QUIC's connection migration.

- Use the same packet header formats as QUIC version 1 to minimize the difference between multipath and non-multipath traffic being exposed on wire.

- Congestion Control must be per-path (following [QUIC-TRANSPORT]) which usually also requires per-path RTT measurements

- PMTU discovery should be performed per-path

- The use of this multipath extension requires the use of non-zero Connection IDs in both directions.

- A path is determined by the 4-tuple of source and destination IP address as well as source and destination port. Therefore, there can be at most one active paths/connection ID per 4-tuple.

- If the 4-tuple changes without the use of a new connection ID (e.g. due to a NAT rebinding), this is considered as a migration event.

The path management specified in Section 9 of [QUIC-TRANSPORT] fulfills multiple goals: it directs a peer to switch sending through a new preferred path, and it allows the peer to release resources associated with the old path. Multipath requires several changes to that mechanism:

- Allow simultaneous transmission of non-probing frames on multiple paths.
- Continue using an existing path even if non-probing frames have been received on another path.
- Manage the removal of paths that have been abandoned.

As such, this extension specifies a departure from the specification of path management in Section 9 of [QUIC-TRANSPORT] and therefore requires negotiation between the two endpoints using a new transport parameter, as specified in Section 3.

This extension uses multiple packet number spaces. When multipath is negotiated, each separate packet number space is linked to a path ID. Using multiple packet number spaces enables direct use of the loss recovery and congestion control mechanisms defined in [QUIC-RECOVERY].

This specification requires the sender to use a non-zero connection ID when opening an additional path. Some deployments of QUIC use zero-length connection IDs. However, when a node selects to use zero-length connection IDs, it is not possible to use different connection IDs for distinguishing packets sent to that node over different paths.

Each endhost may use several IP addresses to serve the connection. In particular, the multipath extension supports the following scenarios.

- The client uses multiple IP addresses and the server listens on only one.
- The client uses only one IP address and the server listens on several ones.
- The client uses multiple IP addresses and the server listens on several ones.
- The client uses only one IP address and the server listens on only one.

Note that in the last scenario, it still remains possible to have multiple paths over the connection, given that a path is not only

defined by the IP addresses being used, but also the port numbers. In particular, the client can use one or several ports per IP address and the server can listen on one or several ports per IP address.

This proposal does not cover address discovery and management. Addresses and the actual decision process to setup or tear down paths are assumed to be handled by the application that is using the QUIC multipath extension. This is sufficient to address the first aforementioned scenario. However, this document does not prevent future extensions from defining mechanisms to address the remaining scenarios. Further, this proposal only specifies a simple basic packet scheduling algorithm, in order to provide some basic implementation guidance. However, more advanced algorithms as well as potential extensions to enhance signaling of the current path state are expected as future work.

1.1. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

We assume that the reader is familiar with the terminology used in [QUIC-TRANSPORT]. When this document uses the term "path", it refers to the notion of "network path" used in [QUIC-TRANSPORT]. In addition, we define the following term:

* Path Identifier (Path ID): An identifier that is used to identify a path in a QUIC connection at an endpoint. Path Identifier is used in multipath control frames (etc. PATH_ABANDON frame) to identify a path. Connection IDs are issued per path ID. When endpoints address a path in multipath control frames, it refers to the Path Identifier related to the destination Connection ID used for sending packets on that particular path.

2. High-level overview

The multipath extensions to QUIC proposed in this document enable the simultaneous utilization of different paths to exchange non-probing QUIC frames for a single connection. This contrasts with the base QUIC protocol [QUIC-TRANSPORT] that includes a connection migration mechanism that selects only one path to exchange such frames.

A multipath QUIC connection starts with a QUIC handshake as a regular QUIC connection. See further Section 3. The peers use the initial_max_paths transport parameter during the handshake to

negotiate the utilization of the multipath capabilities. The initial_max_paths transport parameter limits the initial maximum number of active paths that can be used during a connection. The active_connection_id_limit transport parameter limits the maximum number of active Connection IDs per path. A multipath QUIC connection is thus an established QUIC connection where the initial_max_paths transport parameter has been successfully negotiated.

Endpoints need to pre-allocate new Connection IDs with associating Path Identifiers before initiating new paths. To add a new path to an existing multipath QUIC connection, a client starts a path validation on the chosen path, as further described in Section 5. In this version of the document, a QUIC server does not initiate the creation of a path, but it can validate a new path created by a client. A new path can only be used once the associated 4-tuple has been validated by ensuring that the peer is able to receive packets at that address (see Section 8 of [QUIC-TRANSPORT]). The Path Identifier communicated when advertising a Destination Connection ID is used to associate a packet to a packet number space that is used on a valid path. Further, the Path Identifier associated with Destination Connection ID is used as numerical identifier in control frames. E.g. an endpoint sends a PATH_ABANDON frame to request its peer to abandon the path on which the sender uses the Path Identifier contained in the PATH_ABANDON frame.

In addition to these core features, an application using Multipath QUIC will typically need additional algorithms to handle the number of active paths and how they are used to send packets. As these differ depending on the application's requirements, their specification is out of scope of this document.

Using multiple packet number spaces requires changes in the way AEAD is applied for packet protection, as explained in Section 6.2, and tighter constraints for key updates, as explained in Section 6.3.

3. Handshake Negotiation and Transport Parameter

This extension defines a new transport parameter, used to negotiate the use of the multipath extension during the connection handshake, as specified in [QUIC-TRANSPORT]. The new transport parameter is defined as follows:

* initial_max_paths (current version uses 0x0f739bbc1b666d07): the initial_max_paths transport parameter is included if the endpoint supports the multipath extension as defined in this document. This is a variable-length integer value specifying the maximum number of active concurrent paths an endpoint is willing to build. The value of the initial_max_paths parameter MUST be at

least 2. An endpoint that receives a value less than 2 MUST close the connection with an error of type TRANSPORT_PARAMETER_ERROR. Setting this parameter is equivalent to sending a MAX_PATHS (Section 9.7) of the corresponding type with the same value.

If any of the endpoints does not advertise the initial_max_paths transport parameter, then the endpoints MUST NOT use any frame or mechanism defined in this document.

When advertising the initial_max_paths transport parameter, the endpoint MUST use non-zero source and destination connection IDs. If an initial_max_paths transport parameter is received and the carrying packet contains a zero length connection ID, the receiver MUST treat this as a connection error of type MP_PROTOCOL_VIOLATION and close the connection.

The initial_max_paths parameter MUST NOT be remembered (Section 7.4.1 of [QUIC-TRANSPORT]). New paths can only be used after handshake completion.

This extension does not change the definition of any transport parameter defined in Section 18.2. of [QUIC-TRANSPORT].

The transport parameter "active_connection_id_limit" [QUIC-TRANSPORT] limits the number of usable Connection IDs per path when the initial_max_paths parameter is negotiated successfully. Endpoints might prefer to retain spare Connection IDs so that they can respond to unintentional migration events (Section 9.5 of [QUIC-TRANSPORT]).

Endpoints SHOULD use MP_NEW_CONNECTION_ID and MP_RETIRE_CONNECTION_ID frames to provide new Connection IDs for the peer after the initial_max_paths parameter is negotiated.

Endpoints MUST NOT issue Connection IDs with Path Identifiers larger than the path limitation declared by the initial_max_paths transport parameter and MAX_PATHS frames.

Cipher suites with nonce shorter than 12 bytes cannot be used together with the multipath extension. If such cipher suite is selected and the use of the multipath extension is negotiated, endpoints MUST abort the handshake with a TRANSPORT_PARAMETER error.

4. Path Identifier

The explicit Path Identifier is an integer between 0 and 2^32 - 1 (inclusive). The Path Identifier is pre-allocated when endpoints provide new Connection IDs with MP_NEW_CONNECTION_ID frames Section 9.5.

Each Connection ID is associated with a Path Identifier, as documented in Section 9.5. Multiple connection IDs can be associated with the same path identifier.

Endpoints use Path Identifier to address a path in the multipath control frames, such as PATH_ABANDON, PATH_STANDBY, and PATH_AVAILABLE frames.

Path IDs are generated monotonically increasing, which means the retired Path IDs MUST NOT be reused. Once a Path ID is retired by PATH_ABANDON, it MUST NOT be reused on any other path.

Each endpoint associates a Receiver Packet Number space to each Path Identifier that it provides to the peer. Each endpoint associates a Sender Packet Number space to each Path Identifier received from the peer.

The Path Identifier associated with the Destination Connection ID is used to construct the packet protection nonce defined in {#multipath-aead}.

The Path Identifier associated with the Destination Connection ID is used to identify the path in ACK_MP frames {#ack-mp-frame}.

Note that the Path Identifier for the initial path is 0. Connection IDs which are issued by origin NEW_CONNECTION_ID frames Section 19.15. of [QUIC-TRANSPORT] MUST be treated as their Path Identifier is 0. Also, the Path Identifier for the connection ID specified in the "preferred address" transport parameter is 0. Use of the "preferred address" is considered as a migration event that does not change the path ID.

Endpoints use PATH_ABANDON frame to inform the peer of the retirement of associated Path Identifier. When there is not enough unused Path Identifiers, endpoints SHOULD send MAX_PATHS frame to inform the peer that new Path Identifiers are available.

5. Path Setup and Removal

After completing the handshake, endpoints have agreed to enable multipath feature. They can also start using multiple paths, unless both server preferred addresses and a disable_active_migration transport parameter were provided by the server, in which case a client is forbidden to establish new paths until "after a client has acted on a preferred_address transport parameter" (Section 18.2. of [QUIC-TRANSPORT]).

This document does not specify how an endpoint that is reachable via several addresses announces these addresses to the other endpoint. In particular, if the server uses the preferred_address transport

parameter, clients cannot assume that the initial server address and the addresses contained in this parameter can be simultaneously used for multipath (Section 9.6.2 of [QUIC-TRANSPORT]). Furthermore, this document does not discuss when a client decides to initiate a new path. We delegate such discussion in separate documents.

To let the peer open a new path, an endpoint needs to provide its peer with connection IDs and associated Path Identifiers for the new path.

To open a new path, an endpoint SHALL use different Connection IDs on different paths. Still, the receiver may observe the same Connection ID used on different 4-tuples due to, e.g., NAT rebinding. In such case, the receiver reacts as specified in Section 9.3 of [QUIC-TRANSPORT] by initiating path validation and using a new Connection ID for the same path ID.

This proposal adds three multipath control frames for path management:

*PATH_ABANDON frame for the receiver side to abandon the path (see Section 9.2)

*PATH_STANDBY and PATH_AVAILABLE frames to express a preference in path usage (see Section 9.3 and Section 9.4

All new frames are sent in 1-RTT packets [QUIC-TRANSPORT].

5.1. Path Initiation

Connection IDs cannot be reused, thus opening a new path requires the use of a new Connection ID (see Section 9.5 of [QUIC-TRANSPORT]). Following [QUIC-TRANSPORT], each endpoint uses MP_NEW_CONNECTION_ID frames to issue usable connections IDs to reach it. As such to open a new path by initiating path validation, both sides need at least one Connection ID (see Section 5.1.1 of [QUIC-TRANSPORT]), which is associated with an unused Path ID.

If the transport parameter "initial_max_paths" is negotiated as N, and the client is already actively using N paths, the limit is reached. If the client wants to start a new path, it has to retire one of the established paths.

When the multipath option is negotiated, clients that want to use an additional path MUST first initiate the Address Validation procedure with PATH_CHALLENGE and PATH_RESPONSE frames described in Section 8.2 of [QUIC-TRANSPORT], unless it has previously validated that address. After receiving packets from the client on a new path, if the server decides to use the new path, the server MUST perform

path validation (Section 8.2 of [QUIC-TRANSPORT]) unless it has previously validated that address.

If validation succeeds, the client can continue to use the path. If validation fails, the client MUST NOT use the path and can remove any status associated to the path initation attempt. Section 9.1 of [QUIC-TRANSPORT] introduces the concept of "probing" and "non-probing" frames. When the multipath extension is negotiated, the reception of "non-probing" packet on a new path needs to be considered as a path initiation attempt that does not impact the path status of any existing path. Therefore, any frame can be sent on a new path at any time as long as the anti-amplification limits (Section 21.1.1.1 of [QUIC-TRANSPORT]) and the congestion control limits for this path are respected.

Further, in contrast with the specification in Section 9 of [QUIC-TRANSPORT], the server MUST NOT assume that receiving non-probing packets on a new path with a new Connection ID indicates an attempt to migrate to that path. Instead, servers SHOULD consider new paths over which non-probing packets have been received as available for transmission. Reception of QUIC packets on a new path containing a Connection ID that is already in use on another path should be considered as a path migration as further discussed in Section 8.8.

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. In situations where multiple paths are activated, the client may be recipient of several tokens, each tied to a different address. When considering using a token for subsequent connections, the client ought to carefully select the token to use, due to the inherent ambiguity associated with determining the exact address to which a token is bound. To alleviate such a token ambiguity issue, a server may issue a token that is 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].

5.2. Path State Management

An endpoint uses PATH_STANDBY and PATH_AVAILABLE frames to inform that the peer should send packets in the preference expressed by these frames. Note that the endpoint might not follow the peer’s advertisements, but these frames are still a clear signal of suggestion for the preference of path usage by the peer. Each peer indicates its preference of path usage independently of the other peer. It means that peers may have different usage preferences for the same path. Depending on the sender's decisions, this may lead to

usage of paths that have been indicated as "standby" by the peer or non-usage of some locally available paths.

PATH_AVAILABLE indicates that a path is "available", i.e., it suggests to the peer to use its own logic to split traffic among available paths. PATH_STANDBY marks a path as "standby", i.e., it suggests that no traffic should be sent on that path if another path is available. If no frame indicating a path usage preference was received for a certain path, the preference of the peer is unknown and the sender needs to decide based on it own local logic if the path should be used.

Endpoints use Path Identifier in these frames to identify which path state is going to be changed. Notice that both frames can be sent via a different path and therefore might arrive in different orders. The PATH_AVAILABLE and PATH_STANDBY frames share a common sequence number space to detect and ignore outdated information.

If all available path are marked as "standby", no guidance is provided about which path should be used preferably.

5.3. Path Close

Each endpoint manages the set of paths that are available for transmission. At any time in the connection, each endpoint can decide to abandon one of these paths, following for example changes in local connectivity or changes in local preferences. After an endpoint abandons a path, the peer will not receive any more non-probing packets on that path. Non-probing packets are defined in Section 9.1 of [QUIC-TRANSPORT].

An endpoint that wants to close a path SHOULD use explicit request to terminate the path by sending the PATH_ABANDON frame (see Section 5.3.1). Note that while abandoning a path will cause Connection ID retirement, only retiring the associated Connection ID does not necessarily advertise path abandon (see Section 5.3.4). However, implicit signals such as idle time or packet losses might be the only way for an endhost to detect path closure (see Section 5.3.5).

Note that other explicit closing mechanisms of [QUIC-TRANSPORT] still apply on the whole connection. In particular, the reception of either a CONNECTION_CLOSE (Section 10.2 of [QUIC-TRANSPORT]) or a Stateless Reset (Section 10.3 of [QUIC-TRANSPORT]) closes the connection.

5.3.1. Use PATH_ABANDON Frame to Close a Path

Both endpoints, namely the client and the server, can initiate path closure, by sending a PATH_ABANDON frame (see Section 9.2) which

requests the peer to stop sending packets with the corresponding Path Identifier.

When sending or receiving a PATH_ABANDON frame, endpoints SHOULD wait for at least three times the current Probe Timeout (PTO) interval as defined in Section 6.2 of [QUIC-RECOVERY] after the last packet was sent on the path, before sending the MP_RETIRE_CONNECTION_ID frame for all the corresponding Connection IDs used for this path. This is inline with the requirement of Section 10.2 of [QUIC-TRANSPORT] to ensure that paths close cleanly and that delayed or reordered packets are properly discarded. The effect of receiving a MP_RETIRE_CONNECTION_ID frame is specified in the next section.

Usually, it is expected that the PATH_ABANDON frame is used by the client to indicate to the server that path conditions have changed such that the path is or will be not usable anymore, e.g. in case of a mobility event. The PATH_ABANDON frame therefore recommends to the receiver that no packets should be sent on that path anymore. In addition, the MP_RETIRE_CONNECTION_ID frame is used indicate to the receiving peer that the sender will not send any packets associated to the Connection ID used on that path anymore. The receiver of a PATH_ABANDON frame MAY also send a PATH_ABANDON frame to indicate its own unwillingness to receive any packet on this path anymore.

PATH_ABANDON frames can be sent on any path, not only the path that is intended to be closed. Thus, a path can be abandoned even if connectivity on that path is already broken. Respectively, if there is still an active path, it is RECOMMENDED to send a PATH_ABANDON frame after an idle time on another path.

If a PATH_ABANDON frame is received for the only active path of a QUIC connection, the receiving peer SHOULD send a CONNECTION_CLOSE frame and enter the closing state. If the client received a PATH_ABANDON frame for the last open path, it MAY instead try to open a new path, if available, and only initiate connection closure if path validation fails or a CONNECTION_CLOSE frame is received from the server. Similarly the server MAY wait for a short, limited time such as one PTO if a path probing packet is received on a new path before sending the CONNECTION_CLOSE frame.

Note that PATH_ABANDON frame is also used as a signal for the retirement of the associated Path Identifier. When endpoint received PATH_ABANDON frame, it SHOULD not use the associated Path Identifier in future packets, it can only use the Path ID in ACK_MP frames for inflight packets or in MP_RETIRE_CONNECTION_ID frames for CID retirement.

5.3.2. Refusing a New Path

An endpoint may deny the establishment of a new path initiated by its peer during the address validation procedure. According to [QUIC-TRANSPORT], the standard way to deny the establishment of a path is to not send a PATH_RESPONSE in response to the peer's PATH_CHALLENGE.

A failed path validation consumes the Path ID used for probing of this path. An endpoint MUST not use the same Path ID to probe a different path. Instead, it MUST send a PATH_ABANDON frame to retire the Path ID.

5.3.3. Allocating, Consuming and Retiring Connection IDs

Each endpoints pre-allocate a Path Identifier for each new Connection ID. The Path Identifier 0 indicates the initial path of the connection. Endpoints SHOULD issue at least one unused Connection ID with unused Path Identifier.

An endpoint maintains a set of connection IDs received from its peer for each path, any of which it can use when sending packets, as the same in [QUIC-TRANSPORT]. The difference of multi-path extension is that Connection IDs are pre-allocated for each paths. Each Connection ID is belonging to one path specified by the Path Identifier field of MP_NEW_CONNECTION_ID frame in Section 9.5. The Connection IDs used during handshake are belonging to the initial path with Path Identifier 0.

When the endpoint wishes to remove a connection ID from use, it sends a MP_RETIRE_CONNECTION_ID frame Section 9.6 to its peer. Sending a MP_RETIRE_CONNECTION_ID frame indicates that the connection ID will not be used again. If the path is still active, the peer SHOULD replace it with a new connection ID using a MP_NEW_CONNECTION_ID frame.

Note that Connection Sequeunce number and Retire Prior To field are both used for the corresponding path specified by a Path Identifier.

Upon receipt of an increased Retire Prior To field, the peer MUST stop using the corresponding connection IDs of the specified path and retire them with MP_RETIRE_CONNECTION_ID frames before adding the newly provided connection ID to the set of active connection IDs belonging to the specified path.

Endpoints MUST NOT issue new Connection IDs which has Path Identifiers larger than the max path identifier field in MP_MAX_PATHS frames Section 9.7. When endpoint finds it has not enough available unused Path Identifiers, it SHOULD send a

MP_MAX_PATHS frame to inform the peer that it could use larger active Path Identifiers.

5.3.4. Effect of MP_RETIRE_CONNECTION_ID Frame

Receiving a MP_RETIRE_CONNECTION_ID frame causes an endpoint to discard the resources associated with that Connection ID. Note that retirement of Connection IDs will not effect the use of Path Identifier for the specific path. The list of received packets used to send acknowledgements is also remain uneffected as the packet number space is associated with a path.

The peer, that sent RETIRE_CONNECTION_ID frame, can keep sending data using the same IP addresses and UDP ports previously associated with that Connection ID, but MUST use a different connection ID when doing so. If no new connection ID is available anymore, the endpoint cannot send on this path. This can happen if, e.g., the Connection ID issuer requests retirement of a Connection ID using the Retire Prior To field in the NEW_CONNECTION_ID frame but does provide sufficient new CIDs.

Note that even if a peer cannot send on a path anymore because it does not have a valid Connection ID to use, it can still acknowledge packets received on the path, by sending ACK_MP frames on another path, if available. Also note that, as there is no valid CID associated with the path, both endpoints can still send multipath control frames that contain the Path Identifier on available paths, such as PATH_ABANDON, PATH_STANDBY or PATH_AVAILABLE.

If the peer cannot send on a path and no data is received on the path, the idle time-out will close the path. If, before the idle timer expires, a new Connection ID gets issued by its peer, the endpoint can re-activate the path by sending a packet with a new Connection ID on that path.

If the sender retires a Connection ID that is still used by in-flight packets, it may receive ACK_MP frames referencing the retired Connection ID. If the sender stops tracking sent packets with retired Connection ID, these would be spuriously marked as lost. To avoid such performance issue without keeping retired Connection ID state, an endpoint should first stop sending packets with the to-be-retired Connection ID, then wait for all in-flight packets to be either acknowledged or marked as lost, and finally retire the Connection ID.

5.3.5. Idle Timeout

[QUIC-TRANSPORT] allows for closing of connections if they stay idle for too long. The connection idle timeout in multipath QUIC is defined as "no packet received on any path for the duration of the

idle timeout". When only one path is available, servers MUST follow the specifications in [QUIC-TRANSPORT].

When more than one path is available, hosts shall monitor the arrival of non-probing packets and the acknowledgements for the packets sent over each path. Hosts SHOULD stop sending traffic on a path if for at least the period of the idle timeout as specified in Section 10.1. of [QUIC-TRANSPORT] (a) no non-probing packet was received or (b) no non-probing packet sent over this path was acknowledged, but MAY ignore that rule if it would disqualify all available paths. To avoid idle timeout of a path, endpoints can send ack-eliciting packets such as packets containing PING frames (Section 19.2 of [QUIC-TRANSPORT]) on that path to keep it alive. Sending periodic PING frames also helps prevent middlebox timeout, as discussed in Section 10.1.2 of [QUIC-TRANSPORT].

Server MAY release the resource associated with paths for which no non-probing packet was received for a sufficiently long path-idle delay, but SHOULD only release resource for the last available path if no traffic is received for the duration of the idle timeout, as specified in Section 10.1 of [QUIC-TRANSPORT]. This means if all paths remain idle for the idle timeout, the connection is implicitly closed.

Server implementations need to select the sub-path idle timeout as a trade- off between keeping resources, such as connection IDs, in use for an excessive time or having to promptly reestablish a path after a spurious estimate of path abandonment by the client.

5.4. Path States

Figure 1 shows the states that an endpoint's path can have.

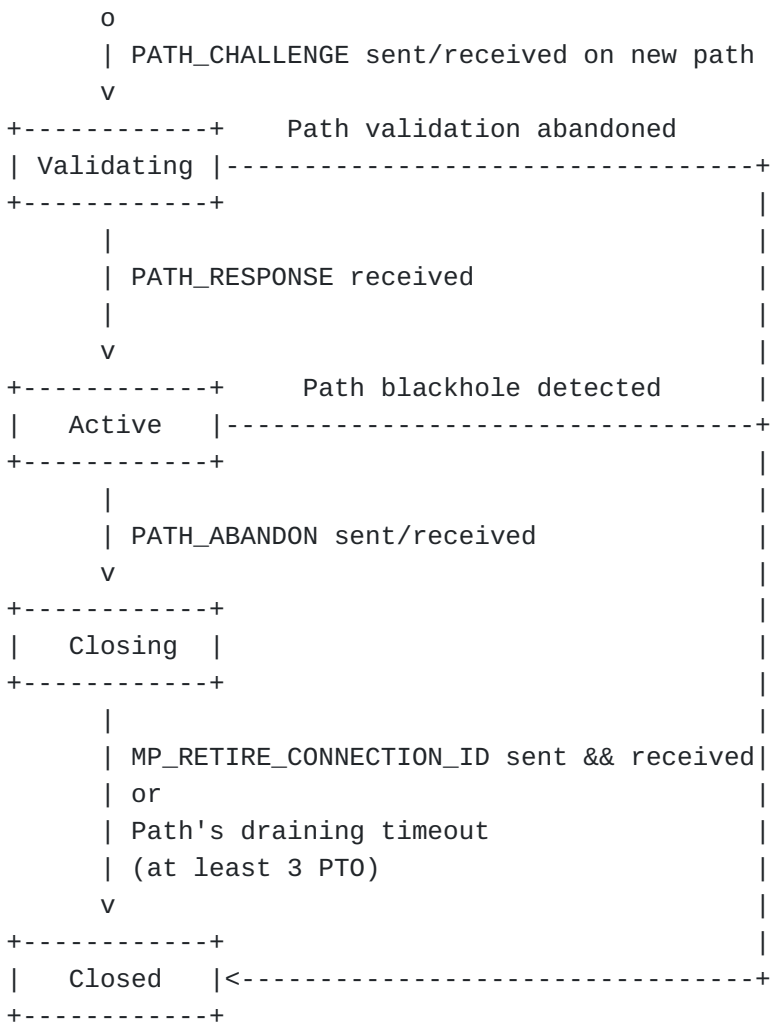

Figure 1: States of a path

In non-final states, hosts have to track the following information.

*Associated 4-tuple: The tuple (source IP, source port, destination IP, destination port) used by the endhost to send packets over the path.

*Associated Path Identifier: The Path Identifier used to address the path. The endpoint relies on its sequence number to send path control information and specifically acknowledge packets belonging to that path-specific packet number space.

*Associated Destination Connection ID: The Connection ID used to send packets over the path.

In Active state, hosts MUST also track the following information:

*Associated Source Connection ID: The Connection ID used to receive packets over the path.

A path in the "Validating" state performs path validation as described in Section 8.2 of [QUIC-TRANSPORT].

The endhost can use all the paths in the "Active" state, provided that the congestion control and flow control currently allow sending of new data on a path. Note that if a path became idle due to a timeout, endpoints SHOULD send PATH_ABANDON frame before closing the path.

In the "Closing" state, the endhost SHOULD NOT send packets on this path anymore, as there is no guarantee that the peer can still map the packets to the connection. The endhost SHOULD wait for the acknowledgment of the PATH_ABANDON frame before moving the path to the "Closed" state to ensure a graceful termination of the path.

When a path reaches the "Closed" state, the endhost releases all the path's associated resources, including the associated Connection IDs. Endpoints SHOULD send MP_RETIRE_CONNECTION_ID frames for releasing the associated Connection IDs following [QUIC-TRANSPORT]. Considering endpoints are not expected to send packets on the current path in the "Closed" state, endpoints can send MP_RETIRE_CONNECTION_ID frames on other available paths. Consequently, the endhost is not able to send nor receive packets on this path anymore.

6. Multipath Operation with Multiple Packet Number Spaces

The QUIC multipath extension uses different packet number spaces for each path. This also means that the same packet number can occur on each path and the packet number is not a unique identifier anymore. This requires changes to the ACK frame as well as packet protection as described in the following subsections.

When multipath is negotiated, each Path Identifier is linked to a separate packet number space. Each PathID-specific packet number space starts at packet number 0. When following the packet number encoding algorithm described in Appendix A.2 of [QUIC-TRANSPORT], the largest packet number (largest_acked) that has been acknowledged by the peer in this new CID's packet number space is initially set to "None".

6.1. Sending Acknowledgements

The ACK_MP frame, as specified in Section 9.1, is used to acknowledge 1-RTT packets. Compared to the QUIC version 1 ACK frame, the ACK_MP frame additionally contains the receiver's Path Identifier associated with the Destination Connection ID to distinguish the path-specific packet number space.

Acknowledgements of Initial and Handshake packets MUST be carried using ACK frames, as specified in [QUIC-TRANSPORT]. The ACK frames, as defined in [QUIC-TRANSPORT], do not carry the Destination Connection ID Path Identifier field to identify the packet number space. If the multipath extension has been successfully negotiated, ACK frames in 1-RTT packets acknowledge packets sent with the Connection ID having path identifier 0.

As soon as the negotiation of multipath support is completed, endpoints SHOULD use ACK_MP frames instead of ACK frames to acknowledge application data packets, including 0-RTT packets, using the initial Connection ID with path identifier 0 after the handshake concluded.

ACK_MP frames (defined in Section 9.1) can be returned on any path. If the ACK_MP is preferred to be sent on the same path as the acknowledged packet (see Section 8.3 for further guidance), it can be beneficial to bundle an ACK_MP frame with the PATH_RESPONSE frame during path validation.

6.2. Packet Protection

Packet protection for QUIC version 1 is specified in Section 5 of [QUIC-TLS]. The general principles of packet protection are not changed for QUIC Multipath. No changes are needed for setting packet protection keys, initial secrets, header protection, use of 0-RTT keys, receiving out-of-order protected packets, receiving protected packets, or retry packet integrity. However, the use of multiple number spaces for 1-RTT packets requires changes in AEAD usage.

Section 5.3 of [QUIC-TLS] specifies AEAD usage, and in particular the use of a nonce, N, formed by combining the packet protection IV with the packet number. When multiple packet number spaces are used, the packet number alone would not guarantee the uniqueness of the nonce.

In order to guarantee the uniqueness of the nonce, the nonce N 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.

Section 9.5 encodes the Path Identifier for Connection IDs as a variable-length integer, allowing values up to 2^32-1; in this specification, a range of less than 2^32-1 values MUST be used before updating the packet protection key.

To calculate the nonce, a 96 bit path-and-packet-number is composed of the least significant 32 bits of the Path Identifier in network byte order, two zero bits, and the 62 bits of the reconstructed QUIC packet number in network byte order. If the IV is larger than 96

bits, the path-and-packet-number is left-padded with zeros to the size of the IV. The exclusive OR of the padded packet number and the IV forms the AEAD nonce.

For example, assuming the IV value is 6b26114b9cba2b63a9e8dd4f, the Path Identifier is 3, and the packet number is aead, the nonce will be set to 6b2611489cba2b63a9e873e2.

Due to the way the nonce is constructed, endpoints MUST NOT use more than 2^32 Connection IDs without a key update.

6.3. Key Update

The Key Phase bit update process for QUIC version 1 is specified in Section 6 of [QUIC-TLS]. The general principles of key update are not changed in this specification. Following QUIC version 1, the Key Phase bit is used to indicate which packet protection keys are used to protect the packet. The Key Phase bit is toggled to signal each subsequent key update.

Because of network delays, packets protected with the older key might arrive later than the packets protected with the new key, however receivers can solely rely on the Key Phase bit to determine the corresponding packet protection key, assuming that there is sufficient interval between two consecutive key updates (Section 6.5 of [QUIC-TLS]).

When this specification is used, endpoints SHOULD wait for at least three times the largest PTO among all the paths before initiating a new key update after receiving an acknowledgement that confirms receipt of the previous key update. This interval is different from that of QUIC version 1 which used three times the PTO of the only one active path.

Following Section 5.4 of [QUIC-TLS], the Key Phase bit is protected, so sending multiple packets with Key Phase bit flipping at the same time should not cause linkability issue.

7. Examples

7.1. Path Establishment

Figure 2 illustrates an example of new path establishment using multiple packet number spaces.

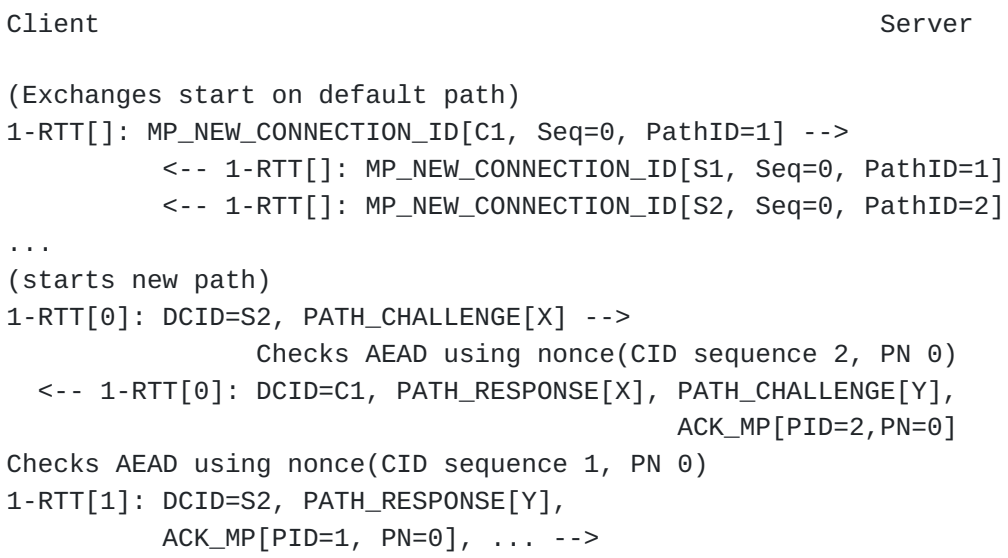

Figure 2: Example of new path establishment

In Figure 2, the endpoints first exchange new available Connection IDs with the NEW_CONNECTION_ID frame. In this example, the client provides one Connection ID (C1 with Path Identifier 1), and server provides two Connection IDs (S1 with Path Identifier 1, and S2 with Path Identifier 2).

Before the client opens a new path by sending a packet on that path with a PATH_CHALLENGE frame, it has to check whether there is an unused Connection IDs available for each side. In this example, the client chooses the Connection ID S2 as the Destination Connection ID in the new path.

If the client has used all the allocated CID, it is supposed to retire those that are not used anymore, and the server is supposed to provide replacements, as specified in [QUIC-TRANSPORT]. Usually, it is desired to provide one more Connection ID as currently in use, to allow for new paths or migration.

7.2. Path Closure

In this example, the client detects the network environment change (client's 4G/Wi-Fi is turned off, Wi-Fi signal is fading to a threshold, or the quality of RTT or loss rate is becoming worse) and wants to close the initial path.

Figure 3 illustrates an example of path closing. For the first path, the server's 1-RTT packets use DCID C1, which has a path identifier of 1; the client's 1-RTT packets use DCID S2, which has a path identifier of 2. For the second path, the server's 1-RTT packets use DCID C2, which has a path identifier of 2; the client's 1-RTT packets use DCID S3, which has a path identifier of 3. Note that the

paths use different packet number spaces. In this case, the client is going to close the first path. It identifies the path by the Path Identifier of the DCID its peer uses for sending packets over that path, hence using the DCID with path identifier 1 (which relates to C1). Optionally, the server confirms the path closure by sending an PATH_ABANDON frame by indicating the path identifier the client uses to send over that path, which corresponds to the path identifier 2 (of S2). Both the client and the server can close the path after receiving the RETIRE_CONNECTION_ID frame for that path.

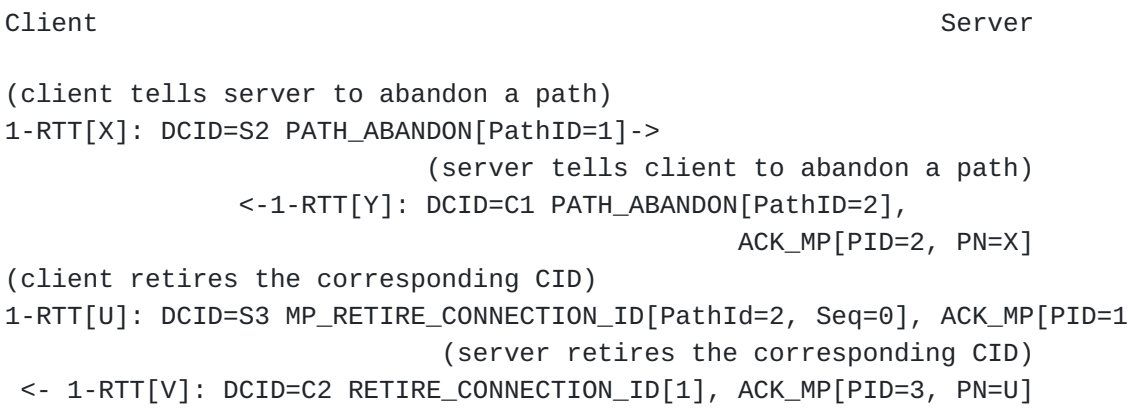

Figure 3: Example of closing a path.

After a path is abandoned, the Path Identifier associated with the path is considered retired and MUST NOT be reused in new paths for security consideration Section 6.2.

Endpoint SHOULD send MAX_PATHS frames Section 9.7 to raise the limit of Path Identifiers when endpoint finds there are not enough unused Path Identifiers (e.g. more than half of the available Path Identifiers are used).

8. Implementation Considerations

8.1. Number Spaces

As stated in Section 1, when multipath is negotiated, each Path Identifier is linked to a separate packet number space. This a big difference between implementations of QUIC as specified in [QUIC-TRANSPORT], which only have to manage three number spaces for Initial, Handshake and Application packets.

Implementation of multipath capable QUIC will need to carefully model the relations between paths and number spaces, as shown in Figure 4.

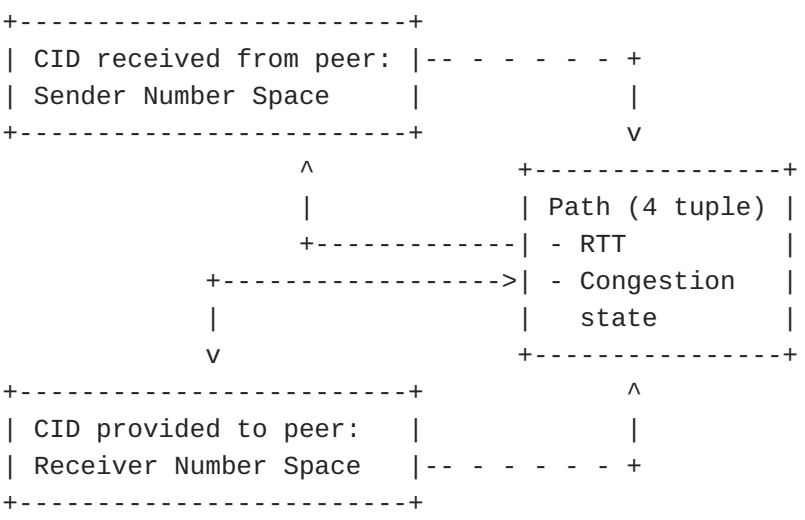

Figure 4: Send and Receive number spaces

The path is defined by the 4-tuple through which packets are received and sent. Packets sent on the path will include the Destination Connection ID currently used for that path, selected from the list of CID provided by the peer. Packets received on the path carry a Destination CID selected by the peer from the list provided to that peer.

The relation between packet number spaces and paths is fixed. CIDs are pre-allocated for each Path Identifier. Once CIDs are issued, they are assigned to one specific Path Identifier. A node may decide to rotate the Destination CID it uses, a NAT may decide to change the 4-tuple over which packets from that path will be received. The packet number space does not change when CID rotation happens within a given Path ID.

Data associated with the transmission and reception on a given path can be associated to either the "path state", or to the state of either the sender or receiver number spaces. For example:

- RTT measurements and congestion state are logically associated with the 4-tuple. They will remain unchanged if data starts being received or sent through the same 4-tuple using new CIDs.

- Implementations of loss recovery typically maintain lists of packets sent and not yet acknowledged. Such information, along with the value of the next PN to use for sending, is logically associated with the "Sender Number Space", which remain unchanged when CID rotation happens.

- Sending of acknowledgement requires keeping track of the PN of received packets and of acknowledgements previously sent. Such

information is logically associated with the "Receiver Number Space", which remain unchanged when CID rotation happens.

8.2. Congestion Control

When the QUIC multipath extension is used, senders manage per-path congestion status as required in Section 9.4 of [QUIC-TRANSPORT]. However, in [QUIC-TRANSPORT] only one active path is assumed and as such the requirement is to reset the congestion control status on path migration. With the multipath extension, multiple paths can be used simultaneously, therefore separate congestion control state is maintained for each path. This means a sender is not allowed to send more data on a given path than congestion control for that path indicates.

When a Multipath QUIC connection uses two or more paths, there is no guarantee that these paths are fully disjoint. When two (or more paths) share the same bottleneck, using a standard congestion control scheme could result in an unfair distribution of the bandwidth with the multipath connection getting more bandwidth than competing single paths connections. Multipath TCP uses the LIA congestion control scheme specified in [RFC6356] to solve this problem. This scheme can immediately be adapted to Multipath QUIC. Other coupled congestion control schemes have been proposed for Multipath TCP such as [OLIA].

8.3. Computing Path RTT

Acknowledgement delays are the sum of two one-way delays, the delay on the packet sending path and the delay on the return path chosen for the acknowledgements. When different paths have different characteristics, this can cause acknowledgement delays to vary widely. Consider for example a multipath transmission using both a terrestrial path, with a latency of 50ms in each direction, and a geostationary satellite path, with a latency of 300ms in both directions. The acknowledgement delay will depend on the combination of paths used for the packet transmission and the ACK transmission, as shown in Table 1.

ACK Path \ Data path	Terrestrial	Satellite
Terrestrial	100ms	350ms
Satellite	350ms	600ms

Table 1: Example of ACK delays using multiple paths

The ACK_MP frames describe packets that were sent on the specified path, but they may be received through any available path. There is an understandable concern that if successive acknowledgements are

received on different paths, the measured RTT samples will fluctuate widely, and that might result in poor performance. While this may be a concern, the actual behavior is complex.

The computed values reflect both the state of the network path and the scheduling decisions by the sender of the ACK_MP frames. In the example above, we may assume that the ACK_MP will be sent over the terrestrial link, because that provides the best response time. In that case, the computed RTT value for the satellite path will be about 350ms. This lower than the 600ms that would be measured if the ACK_MP came over the satellite channel, but it is still the right value for computing for example the PTO timeout: if an ACK_MP is not received after more than 350ms, either the data packet or its ACK_MP were probably lost.

The simplest implementation is to compute smoothedRTT and RTTvar per Section 5.3 of [QUIC-RECOVERY] regardless of the path through which MP_ACKs are received. This algorithm will provide good results, except if the set of paths changes and the ACK_MP sender revisits its sending preferences. This is not very different from what happens on a single path if the routing changes. The RTT, RTT variance and PTO estimates will rapidly converge to reflect the new conditions. There is however an exception: some congestion control functions rely on estimates of the minimum RTT. It might be prudent for nodes to remember the path over which the ACK MP that produced the minimum RTT was received, and to restart the minimum RTT computation if that path is abandoned.

8.4. Packet Scheduling

The transmission of QUIC packets on a regular QUIC connection is regulated by the arrival of data from the application and the congestion control scheme. QUIC packets that increase the number of bytes in flight can only be sent when the congestion window allows it. Multipath QUIC implementations also need to include a packet scheduler that decides, among the paths whose congestion window is open, the path over which the next QUIC packet will be sent. Most frames, including control frames (PATH_CHALLENGE and PATH_RESPONSE being the notable exceptions), can be sent and received on any active path. The scheduling is a local decision, based on the preferences of the application and the implementation.

Note that this implies that an endpoint may send and receive ACK_MP frames on a path different from the one that carried the acknowledged packets. As noted in Section 8.3 the values computed using the standard algorithm reflect both the characteristics of the path and the scheduling algorithm of ACK_MP frames. The estimates will converge faster if the scheduling strategy is stable, but besides that implementations can choose between multiple strategies

such as sending ACK_MP frames on the path they acknowledge packets, or sending ACK_MP frames on the shortest path, which results in shorter control loops and thus better performance.

8.5. Retransmissions

Simultaneous use of multiple paths enables different retransmission strategies to cope with losses such as: a) retransmitting lost frames over the same path, b) retransmitting lost frames on a different or dedicated path, and c) duplicate lost frames on several paths (not recommended for general purpose use due to the network overhead). While this document does not preclude a specific strategy, more detailed specification is out of scope.

8.6. Handling different PMTU sizes

An implementation should take care to handle different PMTU sizes across multiple paths. One simple option if the PMTUs are relatively similar is to apply the minimum PMTU of all paths to each path. The benefit of such an approach is to simplify retransmission processing as the content of lost packets initially sent on one path can be sent on another path without further frame scheduling adaptations.

8.7. Keep Alive

The QUIC specification defines an optional keep alive process, see Section 5.3 of [QUIC-TRANSPORT]. Implementations of the multipath extension should map this keep alive process to a number of paths. Some applications may wish to ensure that one path remains active, while others could prefer to have two or more active paths during the connection lifetime. Different applications will likely require different strategies. Once the implementation has decided which paths to keep alive, it can do so by sending Ping frames on each of these paths before the idle timeout expires.

8.8. Connection ID Changes and NAT Rebindings

Section 5.1.2 of [QUIC-TRANSPORT] indicates that an endpoint can change the Connection ID it uses for to another available one at any time during the connection. As such a sole change of the Connection ID without any change in the address does not indicate a path change and the endpoint can keep the same congestion control and RTT measurement state.

While endpoints assign a Connection ID to a specific sending 4-tuple, networks events such as NAT rebinding may make the packet's receiver observe a different 4-tuple. Servers observing a 4-tuple change will perform path validation (see Section 9 of [QUIC-TRANSPORT]). If path validation process succeeds, the

endpoints set the path's congestion controller and round-trip time estimator according to Section 9.4 of [QUIC-TRANSPORT].

Section 9.3 of [QUIC-TRANSPORT] allows an endpoint to skip validation of a peer address if that address has been seen recently. However, when the multipath extension is used and an endpoint has multiple addresses that could lead to switching between different paths, it should rather maintain multiple open paths instead.

If an endpoint uses a new Connection ID after an idle period and a NAT rebinding leads to a 4-tuple changes on the same packet, the receiving endpoint may not be able to associate the packet to an existing path and will therefore consider this as a new path. This leads to an inconsistent view of open paths at both peers, however, as the "old" path will not work anymore, it will be silently closed after the idle timeout expires.

9. New Frames

All the new frames MUST only be sent in 1-RTT packet.

If an endpoint receives a multipath-specific frame in a different packet type, it MUST close the connection with an error of type FRAME_ENCODING_ERROR.

All multipath-specific frames relate to a Path Identifier of Destination Connection ID. If an endpoint receives a Path Identifier greater than any previously sent to the peer, it MUST treat this as a connection error of type MP_PROTOCOL_VIOLATION. If an endpoint receives a multipath-specific frame with a Path Identifier that it cannot process anymore (e.g., because the path might have been abandoned), it MUST silently ignore the frame.

9.1. ACK_MP Frame

The ACK_MP frame (types TBD-00 and TBD-01) is an extension of the ACK frame defined by [QUIC-TRANSPORT]. It is used to acknowledge packets that were sent on different paths using multiple packet number spaces. If the frame type is TBD-01, ACK_MP frames also contain the sum of QUIC packets with associated ECN marks received on the acknowledged packet number space up to this point.

ACK_MP frame is formatted as shown in Figure 5.

```
ACK_MP Frame {
  Type (i) = TBD-00..TBD-01
       (experiments use  0x15228c00-0x15228c01),
  Path Identifier (i),
  Largest Acknowledged (i),
  ACK Delay (i),
  ACK Range Count (i),
  First ACK Range (i),
  ACK Range (..) ...,
  [ECN Counts (..)],
}
```

Figure 5: ACK_MP Frame Format

Compared to the ACK frame specified in [QUIC-TRANSPORT], the following field is added.

Path Identifier: The path identifier pre-allocated of the Destination Connection ID. This field identifies the packet number space of the 0-RTT and 1-RTT packets which are acknowledged by the ACK_MP frame.

9.2. PATH_ABANDON Frame

The PATH_ABANDON frame informs the peer to abandon a path and retire the Path ID associated.

PATH_ABANDON frames are formatted as shown in Figure 6.

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

Figure 6: PATH_ABANDON Frame Format

PATH_ABANDON frames contain the following fields:

Path Identifier:
The Path Identifier of the Destination Connection ID used by the receiver of the frame to send packets over the path to abandon.

Error Code: A variable-length integer that indicates the reason for abandoning this path.

Reason Phrase Length: A variable-length integer specifying the length of the reason phrase in bytes. Because an PATH_ABANDON frame cannot be split between packets, any limits on packet size will also limit the space available for a reason phrase.

Reason Phrase: Additional diagnostic information for the closure. This can be zero length if the sender chooses not to give details beyond the Error Code value. This SHOULD be a UTF-8 encoded string [RFC3629], though the frame does not carry information, such as language tags, that would aid comprehension by any entity other than the one that created the text.

PATH_ABANDON frames are ack-eliciting. If a packet containing a PATH_ABANDON frame is considered lost, the peer SHOULD repeat it.

9.3. PATH_STANDBY frame

PATH_STANDBY Frames are used by endpoints to inform the peer about its preference to not use the path associated to the Destination Connection IDs in the frame for sending. PATH_STANDBY frames are formatted as shown in Figure 7.

```
PATH_STANDBY Frame {
  Type (i) = TBD-03 (experiments use 0x15228c07)
  Path Identifier (i),
  Path Status sequence number (i),
}
```

Figure 7: PATH_STANDBY Frame Format

PATH_STANDBY Frames contain the following fields:

Path Identifier: The Path Identifier of the Destination Connection ID used by the receiver of this frame to send packets over the path the status update corresponds to. All Destination Connection IDs that have been issued MAY be specified, even if they are not yet in use over a path.

Path Status sequence number: A variable-length integer specifying the sequence number assigned for this PATH_STANDBY frame. The sequence number space is shared with the PATH_AVAILABLE frame and

the sequence number MUST be monotonically increasing generated by the sender of the PATH_STANDBY frame in the same connection. The receiver of the PATH_STANDBY frame needs to use and compare the sequence numbers separately for each Path Identifier.

Frames may be received out of order. A peer MUST ignore an incoming PATH_STANDBY frame if it previously received another PATH_STANDBY frame or PATH_AVAILABLE for the same Path Identifier with a Path Status sequence number equal to or higher than the Path Status sequence number of the incoming frame.

PATH_STANDBY frames are ack-eliciting. If a packet containing a PATH_STANDBY frame is considered lost, the peer SHOULD resend the frame only if it contains the last status sent for that path -- as indicated by the sequence number.

A PATH_STANDBY frame MAY be bundled with a MP_NEW_CONNECTION_ID frame or a PATH_RESPONSE frame in order to indicate the preferred path usage before or during path initiation.

9.4. PATH_AVAILABLE frame

PATH_AVAILABLE frames are used by endpoints to inform the peer that the path associated to the Destination Connection IDs in the frame is available for sending. PATH_AVAILABLE frames are formatted as shown in Figure 8.

```
PATH_AVAILABLE Frame {
  Type (i) = TBD-03 (experiments use 0x15228c08),
  Path Identifier (i),
  Path Status sequence number (i),
}
```

Figure 8: PATH_AVAILABLE Frame Format

PATH_AVAILABLE frames contain the following fields:

Path Identifier: The Path Identifier of the Destination Connection ID used by the receiver of this frame to send packets over the path the status update corresponds to.

Path Status sequence number: A variable-length integer specifying the sequence number assigned for this PATH_AVAILABLE frame. The sequence number space is shared with the PATH_STANDBY frame and the sequence number MUST be monotonically increasing generated by the sender of the PATH_AVAILABLE frame in the same connection. The receiver of the PATH_AVAILABLE frame needs to use and compare the sequence numbers separately for each Path Identifier.

Frames may be received out of order. A peer MUST ignore an incoming PATH_AVAILABLE frame if it previously received another PATH_AVAILABLE frame or PATH_STANDBY frame for the same Path Identifier with a Path Status sequence number equal to or higher than the Path Status sequence number of the incoming frame.

PATH_AVAILABLE frames are ack-eliciting. If a packet containing a PATH_AVAILABLE frame is considered lost, the peer SHOULD resend the frame only if it contains the last status sent for that path -- as indicated by the sequence number.

A PATH_AVAILABLE frame MAY be bundled with a MP_NEW_CONNECTION_ID frame or a PATH_RESPONSE frame in order to indicate the preferred path usage before or during path initiation.

9.5. MP_NEW_CONNECTION_ID frames

An endpoint sends a MP_NEW_CONNECTION_ID frame (type=0x15228c09) instead of the NEW_CONNECTION_ID frame to provide its peer with alternative connection IDs that can be used to break linkability when migrating connections; see Section 19.15 of [QUIC-TRANSPORT].

MP_NEW_CONNECTION_ID frames are formatted as shown in Figure 9.

```
MP_NEW_CONNECTION_ID Frame {
  Type (i) = 0x15228c09,
  Path Identifier (i),
  Sequence Number (i),
  Retire Prior To (i),
  Length (8),
  Connection ID (8..160),
  Stateless Reset Token (128),
}
```

Figure 9: MP_NEW_CONNECTION_ID Frame Format

MP_NEW_CONNECTION_ID frames contain the following fields:

Path Identifier: A path identifier which is pre allocated when the Connection ID is generated, which means the current Connection ID can only be used on the corresponding path.

Sequence Number: The sequence number assigned to the connection ID by the sender on the path specified by Path Identifier, encoded as a variable-length integer. Note that the sequence number is allocated dependently on each path, which means different Connection IDs on different paths may have the same sequence number value.

Retire Prior To:

A variable-length integer indicating which connection IDs should be retired on the path specified by Path Identifier; see Section 5.3.3.

Length: An 8-bit unsigned integer containing the length of the connection ID. Values less than 1 and greater than 20 are invalid and MUST be treated as a connection error of type FRAME_ENCODING_ERROR.

Connection ID: A connection ID of the specified length.

Stateless Reset Token: A 128-bit value that will be used for a stateless reset when the associated connection ID is used.

The Sequence Number field and Retire Prior To field is allocated for each path independently. The Retire Prior To field indicates which connection IDs should be retired on the corresponding path of Path Identifier.

The Retire Prior To field applies to connection IDs established during connection setup if the Path Identifier is 0 indicating the initial path; see Section 5.3.3. The value in the Retire Prior To field MUST be less than or equal to the value in the Sequence Number field. Receiving a value in the Retire Prior To field that is greater than that in the Sequence Number field MUST be treated as a connection error of type FRAME_ENCODING_ERROR.

Length, Connection ID, Stateless Reset Token fields have exactly the same definition in NEW_CONNECTION_ID frame Section 19.15 of [QUIC-TRANSPORT].

Note that Connection IDs issued in NEW_CONNECTION_ID frames MUST be treated as their Path Identifier is 0. Also the retire prior to field of NEW_CONNECTION_ID frames just effect the Connection IDs of initial path with path ID 0. This machanism is compatible with [QUIC-TRANSPORT].

9.6. MP_RETIRE_CONNECTION_ID frames

An endpoint sends a MP_RETIRE_CONNECTION_ID frame (type=0x15228c0a) instead of RETIRE_CONNECTION_ID frame to indicate that it will no longer use a connection ID that was issued by its peer. This includes the connection ID provided during the handshake. Sending a MP_RETIRE_CONNECTION_ID frame also serves as a request to the peer to send additional connection IDs for future use, unless the path specified by Path Identifier has been abandoned. New connection IDs can be delivered to a peer using the MP_NEW_CONNECTION_ID frame (Section 9.5).

Retiring a connection ID invalidates the stateless reset token associated with that connection ID.

MP_RETIRE_CONNECTION_ID frames are formatted as shown in Figure 10.

```
MP_RETIRE_CONNECTION_ID Frame {
  Type (i) = 0x15228c0a,
  Path Identifier (i),
  Sequence Number (i),
}
```

Figure 10: MP_RETIRE_CONNECTION_ID Frame Format

Path Identifier: A path identifier which is pre allocated when the Connection ID is generated, which means the current Connection ID can only be used on the corresponding path.

Sequence Number: The sequence number assigned to the connection ID by the sender on the path specified by Path Identifier, encoded as a variable-length integer.

9.7. MAX_PATHS frames

A MAX_PATHS frame (type=0x15228c0b) informs the peer of the cumulative number of paths it is permitted to open.

MAX_PATHS frames are formatted as shown in Figure 11.

```
MAX_PATHS Frame {
  Type (i) = 0x15228c0b,
  Maximum Paths (i),
}
```

Figure 11: MAX_PATHS Frame Format

MAX_PATHS frames contain the following field:

Maximum Path Identifier: A count of the cumulative number of path that can be opened over the lifetime of the connection. This value cannot exceed 2^32-1, as it is not possible to encode Path IDs larger than 2^32-1. Receipt of a frame that permits opening of a path with Path Identifier larger than this limit MUST be treated as a connection error of type FRAME_ENCODING_ERROR.

Loss or reordering can cause an endpoint to receive a MAX_PATHS frame with a lower path limit than was previously received. MAX_PATHS frames that do not increase the path limit MUST be ignored.

An endpoint MUST NOT initiate a path with a path ID higher than the Maximum Paths value. An endpoint MUST terminate the a connection with an error of type MP_PROTOCOL_VIOLATION if a peer opens more paths than was permitted.

10. Error Codes

Multipath QUIC transport error codes are 62-bit unsigned integers following [QUIC-TRANSPORT].

This section lists the defined multipath QUIC transport error codes that can be used in a CONNECTION_CLOSE frame with a type of 0x1c. These errors apply to the entire connection.

MP_PROTOCOL_VIOLATION (experiments use 0x1001d76d3ded42f3): An endpoint detected an error with protocol compliance that was not covered by more specific error codes.

11. IANA Considerations

This document defines a new transport parameter for the negotiation of enable multiple paths for QUIC, and three new frame types. The draft defines provisional values for experiments, but we expect IANA to allocate short values if the draft is approved.

The following entry in Table 2 should be added to the "QUIC Transport Parameters" registry under the "QUIC Protocol" heading.

Value	Parameter Name.	Specification
TBD (current version uses 0x0f739bbc1b666d07)	initial_max_paths	Section 3

Table 2: Addition to QUIC Transport Parameters Entries

The following frame types defined in Table 3 should be added to the "QUIC Frame Types" registry under the "QUIC Protocol" heading.

Value	Frame Name	Specification
TBD-00 - TBD-01 (experiments use 0x15228c00-0x15228c01)	ACK_MP	Section 9.1
TBD-02 (experiments use 0x15228c05)	PATH_ABANDON	Section 9.2
TBD-03 (experiments use 0x15228c07)	PATH_STANDBY	Section 9.3
TBD-04 (experiments use 0x15228c08)	PATH_AVAILABLE	Section 9.4
TBD-05 (experiments use 0x15228c09)	MP_NEW_CONNECTION_ID	Section 9.5

Value	Frame Name	Specification
TBD-06 (experiments use 0x15228c0a)	MP_RETIRE_CONNECTION_ID	Section 9.6
TBD-06 (experiments use 0x15228c0b)	MAX_PATHS	Section 9.7

Table 3: Addition to QUIC Frame Types Entries

The following transport error code defined in Table 4 should be added to the "QUIC Transport Error Codes" registry under the "QUIC Protocol" heading.

Value	Code	Description	Specification
TBD (experiments use 0x1001d76d3ded42f3)	MP_PROTOCOL_VIOLATION	Multipath protocol violation	Section 10

Table 4: Error Code for Multipath QUIC

12. Security Considerations

TBD

13. Contributors

This document is a collaboration of authors that combines work from three proposals. Further contributors that were also involved one of the original proposals are:

*Qing An

*Zhenyu Li

14. Acknowledgments

Thanks to Marten Seemann and Kazuho Oku for their thorough reviews and valuable contributions!

15. References

15.1. Normative References

[QUIC-RECOVERY] Iyengar, J., Ed. and I. Swett, Ed., "QUIC Loss Detection and Congestion Control", RFC 9002, DOI 10.17487/RFC9002, May 2021, <https://www.rfc-editor.org/rfc/rfc9002>.

[QUIC-TLS] Thomson, M., Ed. and S. Turner, Ed., "Using TLS to Secure QUIC", RFC 9001, DOI 10.17487/RFC9001, May 2021, <https://www.rfc-editor.org/rfc/rfc9001>.

[QUIC-TRANSPORT] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <https://www.rfc-editor.org/rfc/rfc9000>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, DOI 10.17487/RFC3629, November 2003, <https://www.rfc-editor.org/rfc/rfc3629>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

15.2. Informative References

[OLIA] Khalili, R., Gast, N., Popovic, M., Upadhyay, U., and J. Le Boudec, "MPTCP is not pareto-optimal: performance issues and a possible solution", Proceedings of the 8th international conference on Emerging networking experiments and technologies, ACM , 2012.

[RFC6356] Raiciu, C., Handley, M., and D. Wischik, "Coupled Congestion Control for Multipath Transport Protocols", RFC 6356, DOI 10.17487/RFC6356, October 2011, <https://www.rfc-editor.org/rfc/rfc6356>.

Authors' Addresses

Yanmei Liu (editor)
Alibaba Inc.

Email: miaoji.lym@alibaba-inc.com
Additional contact information:
(editor)
Alibaba Inc.

Yunfei Ma
Uber Technologies Inc.

Email: yunfei.ma@uber.com
Additional contact information:

Uber Technologies Inc.

Quentin De Coninck (editor)
University of Mons (UMONS)

Email: quentin.deconinck@umons.ac.be

Olivier Bonaventure
UCLouvain and Tessares

Email: olivier.bonaventure@uclouvain.be

Christian Huitema
Private Octopus Inc.

Email: huitema@huitema.net

Mirja Kuehlewind (editor)
Ericsson

Email: mirja.kuehlewind@ericsson.com