QUIC
Internet-Draft
Intended status: Standards Track
Expires: April 25, 2021

Q. An
Y. Liu
Y. Ma
Alibaba Inc.
Z. Li
ICT-CAS

October 22, 2020

Multipath Extension for QUIC
draft-an-multipath-quic-00

Abstract

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

The extension is compliant with the single-path QUIC design. The design principle is to support multipath by adding limited extension to QUIC-Transport [I-D.ietf-quic-transport].

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 April 25, 2021.

Copyright Notice

Table of Contents

1. Introduction 3
2. Notational Conventions 3
3. Sub-Connection 4
4. Enable Multipath QUIC - Handshake 5
5. Sub-connection Management 5
5.1. Multipath QUIC Interaction 5
5.2. Path validation and sub-connection ID negotiation 8
5.3. New sub-connection establishment 9
5.4. Close sub-connection 9
5.5. Sub-connection Lookup 10
5.6. Sub-connection migration 10
5.7. Sub-connection state machine management 10
5.8. Sender and Receiver Connection (Sub-connection) States . 11
5.9. Use load balancer in Multipath QUIC 11
6. Packet scheduling 11
6.1. Overview 11
6.2. Basic Static Scheduling Strategy 12
6.3. Dynamic (feedback-based) Scheduling Strategy 13
6.4. Application Policy-Awareness 14
6.4.1. Per-connection Policy 15
6.4.2. Per-stream Policy 17
7. Congestion control and loss detect 18
7.1. Congestion control 18
7.2. Packet number space and acknowledgements 18
7.3. Flow control 18
8. New frames 18
8.1. MP_SUB_CONN_NEW frame 19
8.2. MP_SUB_CONN_ACCEPT frame 19
8.3. MP_SUB_CONN_CLOSE frame 19
8.4. MP_ACK frame 20
8.5. MP_ADD_ADDRESS frame 21
8.6. MP_REMOVE_ADDRESS frame 21
9. IANA Considerations 21
10. Informative References 21
Authors' Addresses 22

1. Introduction

In this document, we propose an extension to the current QUIC design to enable the simultaneous usage of multiple paths for a single connection.

This proposal differs from past proposals [I-D.deconinck-quic-multipath] in two fundamental perspectives:

- The multi-path QUIC is built on top of the concept of the bidirectional sub-connection, which readily fits into the nature of both cellular and wifi links that cover the majority of multi-path applications in QUIC while keeping the design simple and easy to implement. In doing so, we are able to re-use most of the current QUIC transport design with the sole addition of six new frames.

- The multi-path QUIC design enables feedback-based dynamic scheduling strategy. As the major goal of multi-path QUIC is to enhance performance in mobile applications, where the sender and receiver may have different viewpoints about the fast-changing wireless connectivity, especially in high-mobility scenarios, the proposed design allows the sender and receiver to synchronize their viewpoints via message exchange in ACK packet in order to maximize performance.

This document is organized as follows. It first provides definition of sub-connection in Section 3. It then specifies how to enable multipath QUIC during handshake in Section 4, and sub-connection management in Section 5. It discusses packet scheduling in Section 6, and congestion control in Section 7. It specifies the new frames in Section 8.

2. Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

We assume that the reader is familiar with the terminology used in [I-D.ietf-quic-transport]. In addition, we define the following terms:

- Path: A sequence of links between a sender and a receiver, defined in this context by a 4-tuple of source and destination address/ port pairs[RFC8684].

- Sub-Connection: Sub-connection is bidirectional and provides reliable transmission between client and server. A connection can contain one or multiple sub-connections. A sub-connection is identified by an internal identifier, called Sub-Connection Index (SCI). Each sub-connection has its unique Source Connection ID and Destination Connection ID. The Connection ID is mapped with the 2-tuple of IP address and port.

- (Multipath QUIC) Connection: A set of one or more sub-connections, over which an application can communicate between two host.

3. Sub-Connection

A connection can contain one or multiple sub-connections which are bidirectional and provides reliable transmission between client and server. Sub-connection is identified by Sub-Connection Index (SCI).

If a connection contains at least 2 sub-connections, then the first established sub-connection is called Initial sub-connection. The rest sub-connections are called supplementary sub-connections.

Every sub-connection has its own unique CID pair that is associated with the 4-tuple (source IP, source port, destination IP, destination port) of the underlying network path currently used by the sub-connection. The Connection ID negotiation process is specified in Section 5.1. In case of sub-connection migration, the CID pair will be renegotiated following the connection migration procedure specified in [I-D.ietf-quic-transport].

Endpoints can find which sub-connection a received packet belongs to according to the CID pair of the packet. Endpoints can find the context of a sub-connection by its' CID pair or SCI. In the context of a sub-connection, a reference pointer MUST be provided to access the context of the multipath QUIC connection that the sub-connection belongs to.

Each sub-connection has its independent Packet Number Space. And all sub-connections in the same connection share the same 1-RTT encryption key which is generated during the connection's cryptographic handshake.

Note: The reason of using SCI to identify a Sub-connection: acknowledgements may not be transferred via the same sub-connection where the packets were sent, therefore the MP_ACK frame SHOULD contain field that can uniquely identify the sub-connection, and the same logic applies to other new MP frames. If we use Connection ID to identify a sub-connection in MP frames, the length of Connection ID is too long and will add more overhead in the frames.

4. Enable Multipath QUIC - Handshake

The connection handshake flow follows QUIC-Transport [I-D.ietf-quic-transport], using the transport parameter to negotiate multipath feature. The negotiation mechanism is similar to the negotiation in [I-D.deconinck-quic-multipath] Section 5.1, while the semantic of the transport parameter is different.

A new transport parameter is defined:

o name: max_sub_conn_index (0x40)

o value: a variable-length integer (1 to 8 bytes)

The value range and definition:

o 0: MP feature disabled

o [1, 2^31-1]: the maximum number of sub-connections

The value of SCI(sub-connection index) starts from 1 and increases by 1 when a new sub-connection is created. The value range of SCI is [1, max_sub_conn_index]. The SCI of initial sub-connection is 1. A multipath QUIC connection MUST NOT reuse any used SCI for new sub-connections in its' lifetime.

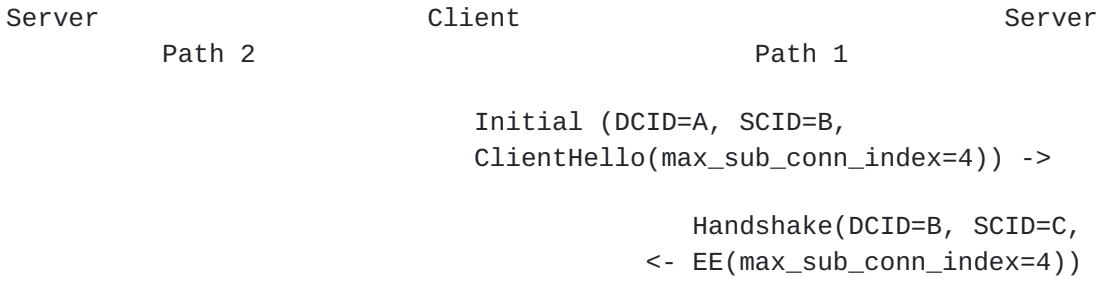
If the peer does not carry the max_sub_conn_index(0x40) transport parameter, which means the peer does NOT support multipath, endpoint MUST fallback to QUIC-Transport [I-D.ietf-quic-transport] with single path, and MUST NOT send any MP frames in the following packets.

5. Sub-connection Management

This section describes the details of sub-connection management.

5.1. Multipath QUIC Interaction

Figure 1 illustrates the Multipath QUIC interaction process.



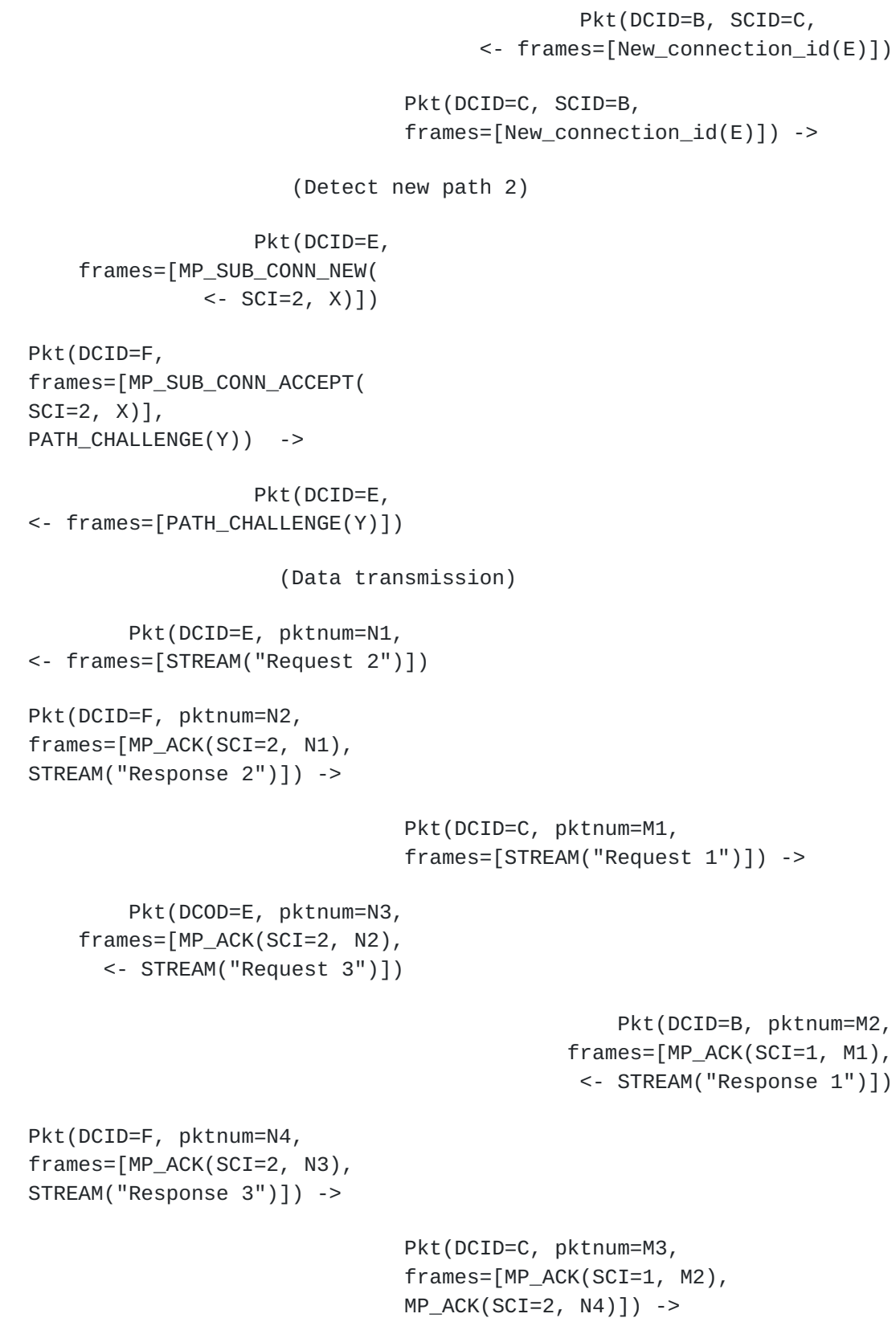

Pkt(DCID=B, SCID=C,
<- frames=[New_connection_id(E)])
Pkt(DCID=C, SCID=B,
frames=[New_connection_id(E)]) ->
(Detect new path 2)
Pkt(DCID=E,
frames=[MP_SUB_CONN_NEW(
<- SCI=2, X)])
Pkt(DCID=F,
frames=[MP_SUB_CONN_ACCEPT(
SCI=2, X)],
PATH_CHALLENGE(Y)) ->
Pkt(DCID=E,
<- frames=[PATH_CHALLENGE(Y)])
(Data transmission)
Pkt(DCID=E, pktnum=N1,
<- frames=[STREAM("Request 2")])
Pkt(DCID=F, pktnum=N2,
frames=[MP_ACK(SCI=2, N1),
STREAM("Response 2")]) ->
Pkt(DCID=C, pktnum=M1,
frames=[STREAM("Request 1")]) ->
Pkt(DCOD=E, pktnum=N3,
frames=[MP_ACK(SCI=2, N2),
<- STREAM("Request 3")])
Pkt(DCID=B, pktnum=M2,
frames=[MP_ACK(SCI=1, M1),
<- STREAM("Response 1")])
Pkt(DCID=F, pktnum=N4,
frames=[MP_ACK(SCI=2, N3),
STREAM("Response 3")]) ->
Pkt(DCID=C, pktnum=M3,
frames=[MP_ACK(SCI=1, M2),
MP_ACK(SCI=2, N4)]) ->

Figure 1: Multipath QUIC interaction process

The process is composed of four phases.

A. Handshake negotiation

During the QUIC-Transport [I-D.ietf-quic-transport] handshake, endpoints negotiate whether multipath feature is supported. The negotiation parameter (see Section 4) is carried within the transport parameters of TLS cryptographic handshake. After the handshake finished, the connection contains the initial sub-connection with SCI equals 1. In Figure 1, the maximum sub-connection index is four.

B. Exchange unused Connection ID in advance

After the two endpoints complete the connection establishment, they can exchange unused Connection IDs by NEW_CONNECTION_ID frame. Before an endpoint starts to create a new sub-connection, it SHOULD check if there are unused Connection IDs for both endpoints.

Note: QUIC-Transport [I-D.ietf-quic-transport] requires Connection ID is uniquely mapped with 2-tuple of IP address and port. If client attempts to use a new 2-tuple as source address to establish a new sub-connection, a new Connection ID is required for client, and also a new Connection ID is required for server.

C. New sub-connection establishment

During this phase, a new sub-connection is established between client and server, and address validation is needed.

When client detects a new network path, it MAY attempts to establish a new sub-connection by sending MP_NEW_SUB_CONN frame which carries a 64-bit random value and claims the new sub-connection's SCI (which is 2 in the example flow in Figure 1). The establishment of sub-connection is always initiated by client.

After the server receives the MP_NEW_SUB_CONN frame from the client, it responds with MP_SUB_CONN_ACCEPT frame which carries the identical 64-bit random value from the received MP_NEW_SUB_CONN frame and agrees with the sub-connection's SCI (2 in the example). Server MUST also perform path validation following the procedure specified in QUIC-Transport [I-D.ietf-quic-transport]. Once the server successfully validates its' peers' address, the new sub-connection is established.

D. Data transmission on new sub-connection

As soon as sub-connections are established, endpoints can communicate with each others over the newly established sub-connections. All valid short header packets defined in QUIC-Transport [I-D.ietf-quic-transport] can be carried on these sub-connections. Every sub-connection has its' independent PNS. Thus, standard QUIC ACK frames defined in QUIC-Transport [I-D.ietf-quic-transport] only acknowledge packets that belong to the same PNS of the sub-connection on which the ACK frames were received.

To enable endpoints reply acknowledgements on different sub-connections rather than the sub-connection where the corresponding packets were received, a new type of frame, MP_ACK, is defined. MP_ACK frames can also be replied over the same sub-connection on which data packets were received. In this case, MP_ACK frames serves very similar purposes as QUIC ACK frames do.

MP_ACK frame contains the sub-connection index of the packets to be acknowledged. For example, in Figure 1, the packet (packet number is N4) is sent via the second sub-connection (SCI is 2), and its corresponding acknowledgement MP_ACK(Sub-Connection Index=2, N4) is sent via the initial sub-connection.

5.2. Path validation and sub-connection ID negotiation

Before clients initiate new sub-connections by sending MP_SUB_CONN_NEW frames to servers through their additional network addresses, they MAY want to validate the reachability between their new network addresses and servers' addresses. In this case, clients can initiate a path validation procedure as specified in QUIC-Transport [I-D.ietf-quic-transport] per address pair.

Path validation uses the PATH_CHALLENGE and PATH_RESPONSE frame defined in QUIC-Transport [I-D.ietf-quic-transport].

Each sub-connection MUST has a unique pair of SCID and DCID within a multipath QUIC connection. Thus, endpoints MUST NOT initiate or accept new sub-connections if they currently have no free CIDs supplied by their peers. In this case, endpoints SHOULD announce new free CIDs to their peers by exchanging NEW_CONNECTION_ID frames.

To ensure that endpoints have free CIDs to create new sub-connections as soon as they get new network addresses, an endpoint SHOULD announce a least one free CID to its peer by sending NEW_CONNECTION_ID frame [I-D.ietf-quic-transport] over its initial sub-connection as soon as the handshake on the initial sub-connection is completed. Endpoints MAY also track the number of free CIDs that their peers can use and announce more free CIDs if needed.

Sub-connection ID negotiation follows the Connection ID negotiation method in Connection Migration defined in QUIC-Transport [I-D.ietf-quic-transport], which is to let client and server claim its own unused Connection ID in advance by NEW_CONNECTION_ID frame. If there is no available unused Connection ID, then establishment of new sub-connection is not allowed.

5.3. New sub-connection establishment

New sub-connection establishment is always initiated by client, by sending MP_NEW_SUB_CONN frame.

Because source address(2-tuple of IP address and port) is usually different in the new network path, client needs to generate and claim new Source Connection IDs prior to the new sub-connection establishment.

Client that sends the MP_SUB_CONN_NEW frame in 1-RTT packets with short headers, MUST use the unused Connection ID claimed in advance by server as Destination Connection ID. MP_SUB_CONN_NEW frame carries a 64-bit random value, and a SCI (increased progressively).

After receiving the MP_SUB_CONN_NEW frame, server responds with MP_SUB_CONN_ACCEPT frame carrying the identical SCI and identical 64-bit random value from the received MP_NEW_SUB_CONN frame. Then, server sends PATH_CHALLENGE to verify the client address.

After client receives the PATH_CHALLENGE frame, it replies with PATH_RESPONSE frame In the following 1-RTT packet (short header) to complete the address validation. After the address validation is completed, client and server can send and receive data unrestrictedly on the established sub-connection.

Before the client's address validation is completed, server needs to limit the cumulative size of packets it sends to an unvalidated address to three times the size of packets it receives from that address in the new sub-connection (to prevent amplification attack).

5.4. Close sub-connection

Both client and server can terminate a sub-connection, by sending MP_SUB_CONN_CLOSE frame that carries a SCI. In scenarios such as client detects the network environment change (client's 4G/Wi-Fi is turned off, Wi-Fi signal is fading to a threshold), or endpoints detect that the quality of RTT or loss rate is becoming worse, client or server can terminate a sub-connection immediately.

MP_SUB_CONN_CLOSE frame can be sent via a different sub-connection instead of the sub-connection to be closed.

5.5. Sub-connection Lookup

Endpoints use Connection IDs to find the context of a connection. Figure 2 illustrates the Connection context. Each sub-connection's Connection IDs can be mapped to the connection.

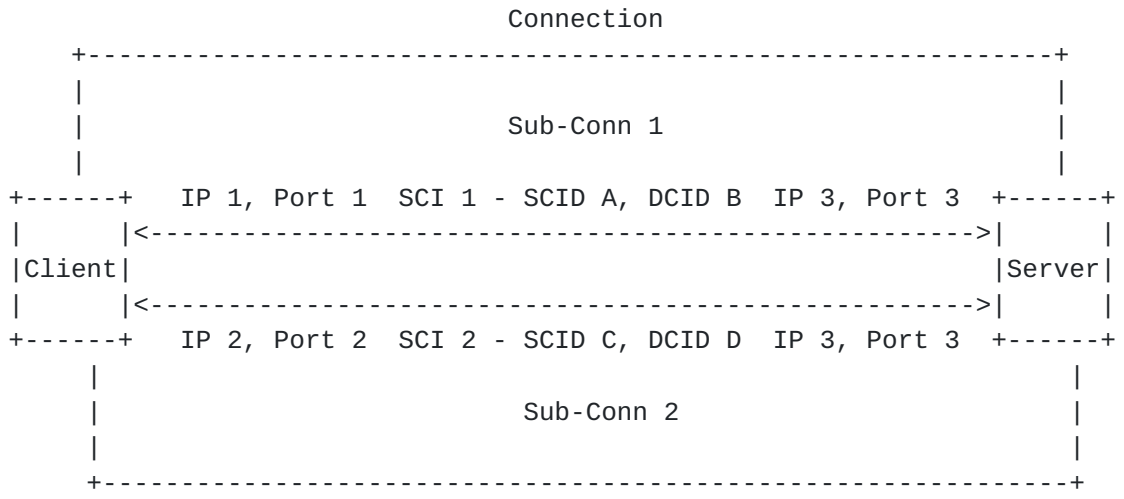


Figure 2: Connection context

In the connection context, client and server can use SCI or Connection ID to find a sub-connection. Note that if sub-connection migration happens, sub-connection's Connection ID need to be renegotiated (See Section 5.6), while the SCI of sub-connection could remain unchanged.

5.6. Sub-connection migration

Sub-connection migration follows the Connection Migration defined in QUIC-Transport [I-D.ietf-quic-transport]. When client experiences NAT rebinding (source address is changed), server needs to revalidate the client address.

5.7. Sub-connection state machine management

TODO

5.8. Sender and Receiver Connection (Sub-connection) States

For each sender and receiver, the sub-connection states include:

Sender	SubConnectionIndex(SCI)	CIDs(SCID, DCID)	4-tuple(sIP, dIP, sPort, dPort)	packet number space
Receiver	SubConnectionIndex(SCI)	CIDs(SCID, DCID)	4-tuple(sIP', dIP', sPort', dPort')	packet number space

Table 1: Sender and Receiver Connection (Sub-connection) States

5.9. Use load balancer in Multipath QUIC

This specification follows the Connection ID negotiation defined in QUIC-Transport [I-D.ietf-quic-transport]. For stateless or low-state load balancers supporting Multipath QUIC, implementations SHOULD use the specification of Connection ID generation and Load balancer routing defined in QUIC-LB [I-D.ietf-quic-load-balancers], guarantee that packets with Connection IDs belonging to the same connection, can be routed to same server.

6. Packet scheduling

6.1. Overview

For an outgoing packet, the packet scheduler decides which sub-connection the packet shall be transmitted. The concept of packet scheduler in Multipath QUIC is similar to that in MPTCP. As long as more than one path's congestion controller allows for a new packet transmission, the packet scheduler is enabled. However, the proposed packet scheduler in this draft differs from past MPTCP proposals in the following aspects:

- We enable dynamic (feedback-based) scheduling strategy with feedback in this proposal to further enhance quality of user experience(QoE) and to facilitate the expression of the application policy-awareness. Such a capability is made available by adding the QoE control signal length field and QoE control signal field in MP_ACK frame (see Section 8.4). With the help of such extension, a receiver is able to interact with a sender's scheduling strategy in real time.

- Unlike MPTCP which send ACK packet over the same path, multipath QUIC allows a packet to be acknowledged over a different path, which allows multipath QUIC to better handle the uplink-downlink heterogeneity in wireless networks.

- We support application policy-awareness in multipath QUIC. An application can implement both per-connection policies and per-stream policies. For example, a live streaming application is allowed to choose a different policy from a web application. The per-connection policy includes path mode, path preference, scheduling algorithm and packet redundancy strategy. A per-stream policy is associated with user-defined stream priorities to express the applications's intent.

6.2. Basic Static Scheduling Strategy

A basic static scheduling strategy consists of four major components:

- Path mode: A scheduler may want to decide which sub-connections shall be activated to transmit data. For instance, a scheduler can choose to use only one of the two sub-connections and completely ignore the other one. A scheduler marks the selected sub-connections to be in the "active" state and the un-selected ones in the "inactive" state.

- Path preference: Due to the fact that costs of transmitting data over different sub-connections are not always equal. For example, the energy (battery) cost over a 5G sub-connection and a Wi-Fi sub-connection are very different, so a user may prefer the Wi-Fi sub-connection when his/her cell phone's battery is low. In another example, transmissions over a Wi-Fi sub-connection and a cellular sub-connection may incur different service charges per packet such that a user prefers to use the Wi-Fi sub-connection over the LTE one. Note that a user's preference may change over time. For instance, certain mobile carriers offer unlimited free data for a particular streaming app. Therefore, the sub-connection priority should be made available in the scheduler.

- Sub-connection selection algorithm：A selection algorithm splits packets across different sub-connections and determines the order of sub-connections to be selected. The selection algorithm takes congestion controller states as inputs, such as smoothed RTTs (sRTTs), estimated bandwidths (eBWs) and congestion window sizes (CWNDs) as well as application-defined information such as sub-connection priorities and path states. The outputs of the algorithm is an ordered list of sub-connections to put a packet on. To name a few, some of the commonly used algorithms are:

- o
  - * Round-Robin: There is no priority. it selects sub-connections one by one in order to transmit data.
  - * Lowest-RTT: It first chooses the sub-connection with the lowest RTT and feeds packets to it until that sub-connection's congestion window is full. Then it chooses the sub-connection with the second lowest RTT.
  - * Highest-Sending-Rate: It first chooses the sub-connection with the highest bandwidth and feeds packets to it until that sub-connection's congestion window is full. Then it chooses sub-connection with the second largest bandwidth.

- o Packet redundancy strategy: One major challenge in multi-path transmission is that a packet loss on the slow sub-connection might block the overall transmission when packets are split across fast-changing sub-connections. As the sub-connection selection algorithm takes inputs from congestion controllers which are basically rough predictions of the network and may not be accurate enough for fast-changing wireless channels, such an imprecise estimation could lead to network overuse/underuse. A solution to this problem is to implement packet redundancy strategy. A redundancy strategy can be applied to only ACK packets (partial redundancy) or all data packets (full redundancy). The multipath QUIC in this draft is designed to enable such flexible redundancy strategies. It is up to the application to determine whether, when, and on which packets to activate transmission redundancy.

6.3. Dynamic (feedback-based) Scheduling Strategy

An important feature of this proposal is the capability of dynamic (feedback-based) scheduling. In a dynamic scheduling strategy, a receiver notifies its currently preferred scheduling strategy to a sender. Such feedback information is carried by QoE control signal in MP_ACK frames. The frequency of such feedback can be controlled to limit the amount of extra information. To do so, four types of MP_ACK frames are designed (Figure 8):

- o Type(i) = 0x22 , with no ECN Counts and no QoE Control Signals
- o Type(i) = 0x23 , with ECN Counts and no QoE Control Signals
- o Type(i) = 0x24 , with no ECN Counts and QoE Control Signals
- o Type(i) = 0x25 , with ECN Counts and QoE Control Signals

The type 0x24 and 0x25 give the flexibility of carrying QoE control signals. Given that the sender and the receiver may have different views of the wireless environments, especially in high-mobility scenarios, the QoE control signal allows a synchronization between their viewpoints dynamically. It is up to the application to determine the interpretation of QoE control signal and its encoding method.

6.4. Application Policy-Awareness

Applications may have completely different QoE requirements---the interactive applications are delay sensitive, while the video streaming applications are more throughput sensitive. There is thus a trend of cross-layer design that tries to take applications' demands into account when managing paths or scheduling packets. The static scheduling strategy and the dynamic scheduling strategy are used together to fully support application policy-awareness in multipath scheduling. To be more specifically, a 'control plane' is separated from a `data plane' as in software-defined networking. The 'control plane' takes applications' high-level demands (a.k.a intent) as input to generate the corresponding policies, which later are deployed on the 'data plane'. The 'data plane' maps users policies to the 'actions', which control the packet scheduler and other functionalities that the transport implements. To allow maximum design flexibility, the proposed multipath QUIC let applications to access/change every single logic of the packet scheduling and path management. The application policy consists of two layers: per-connection policy and per-stream policy.

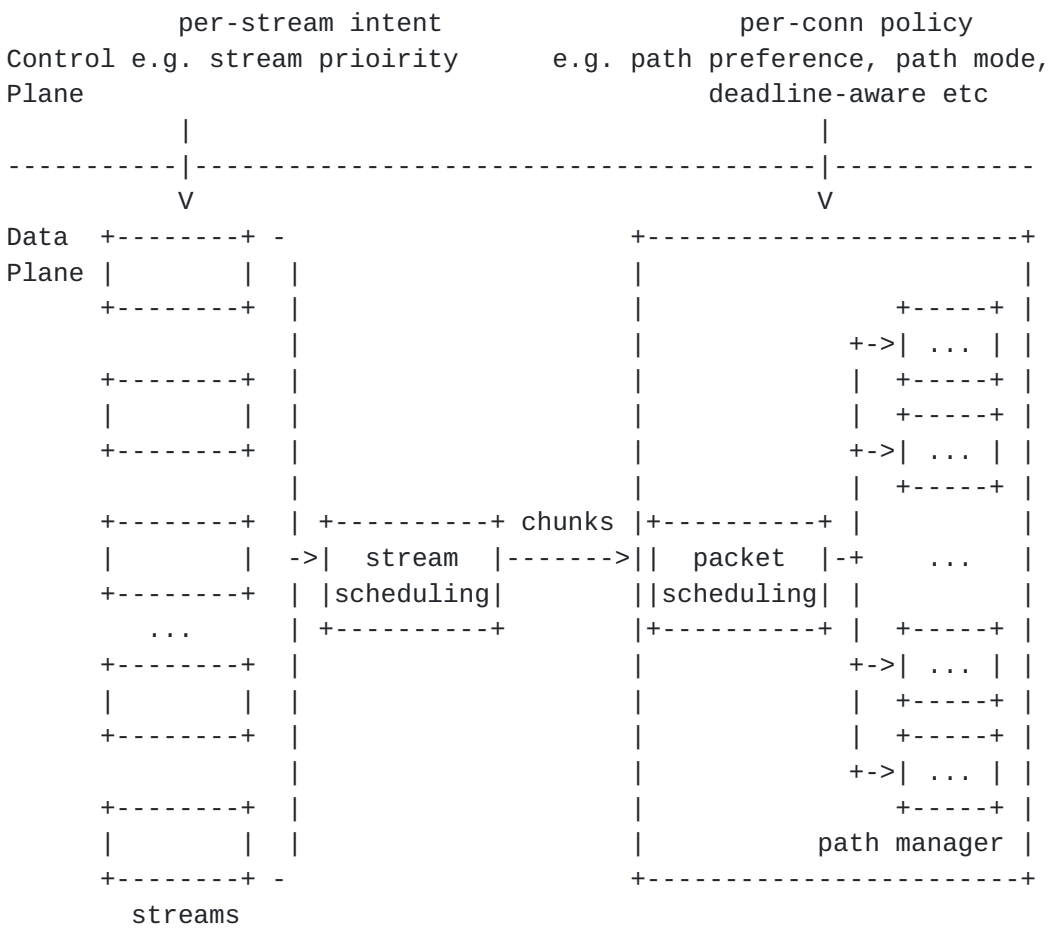


Figure 3: Application Policy Awareness in Multi-path QUIC framework

6.4.1. Per-connection Policy

An application imposes per-connection policy through the primitives provided by the control plane.

As described above, the policy is translated into indications on sub-connection states, sub-connection priorities, sub-connection selection algorithms and packet redundant strategies. The packet scheduler at the data plane will act based on these indications. We assume the policies are 'soft'---the policies are not a must. Instead, the data plane will follow the policies as much as possible.

No.	Application defined policy: Path mode	Application defined policy: Path Preference	Underlying action: Packet Scheduling	Underlying action: Path mngm.
1	Wi-Fi=full, Cellular=full	Wi-Fi=1, Cellular=1	full redundant	/
2	Wi-Fi=full, Cellular=backup	Wi-Fi=1, Cellular=1	full redundant	activate backup interfaces when the active one's performance is lower than X for 5s
3	Wi-Fi=full, Cellular=full	Wi-Fi=2, Cellular=1	partially redundant	/

Table 2: Example policies of a real-time interaction application

Let us take real-time interaction applications as an example to illustrate the basic idea. The applications are indeed delay sensitive but data volume is often low. 3 types of policies may be used by different applications, as shown in Table 2 where we assume only two paths are available (Wi-Fi and Cellular)

The first type of policies would like to use two paths equally, and because the applications are delay sensitive, the actions will be to active 'full redundancy' for the packet redundancy strategy---two paths send the same data. The second type of policies, on the other hand, would like to use the Wi-Fi interface (possibly because of data charge) as much as possible, hence giving the Wi-Fi sub-connection a higher priority. But if two paths have to be activated at the same time due to the lower performance of Wi-Fi, then the two paths are set with same the priority which can be configured dynamically through QoE control signal in MP_ACK feedbacks. The third type of policies would like to use the two interfaces at the same time, but Wi-Fi is preferred twice as the cellular one. The actions will take this into consideration, by implementing a weighed round-robin sub-connection selection algorithm.

Likewise, we can define a mapping between the policies of different types of applications and the actions in the data plane. We leave the design of such a mapping to the designers.

6.4.2. Per-stream Policy

Per-stream intent is a unique feature provided by (MP)QUIC---it is implemented through the multiple streams in QUIC. Streams can be associated with priorities to implement applications intent. For instance, objects in a web page may be dependent on others and thus have different priorities [MPQUIC-Scheduler]. A stream priority-aware packet scheduling algorithm will improve the performance notably.

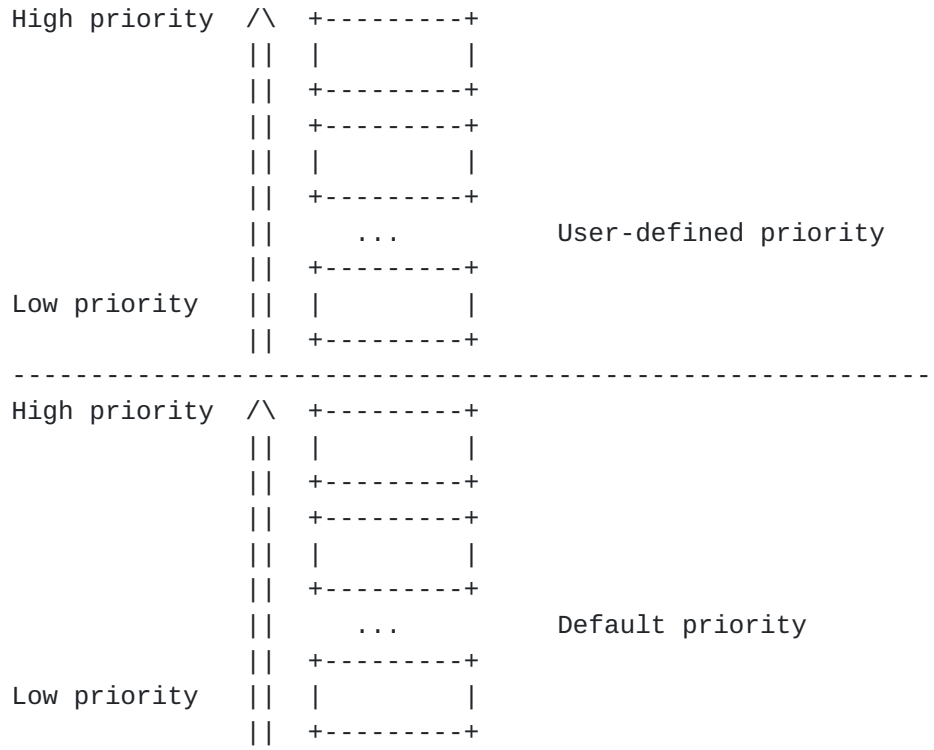


Figure 4: Stream priority

We envision a priority management scheme of two separated priority ranges (see Figure 4). The user-defined priority ranges are those streams that the applications explicitly designate the priorities, where the default priority ranges include the streams with no priority values set by the applications. Only when the streams in the user-defined ranges have no data sent, the data in the streams in the default priority ranges can be sent. In the same range, one can use the weighted round robin for scheduling---the higher-priority

streams get more quantum for data sending in each round. One can also dynamically set/change the priorities of the streams in the default priority ranges to enable short stream first if needed.

7. Congestion control and loss detect

7.1. Congestion control

Implementations MAY support coupled congestion controllers such as LIA [MPTCP-LIA], OLIA [MPTCP-OLIA]s, and etc., or support decoupled congestion controllers in environments using disjoint network paths.

In decoupled congestion control, every sub-connection runs its own congestion controller without interacting with the congestion controllers of other sub-connections. That is to say, in the aspect of congestion control, a sub-connection behaves exactly the same as a normal QUIC connection over the same network path.

Each sub-connection MAY choose congestion control algorithm independently.

7.2. Packet number space and acknowledgements

Every sub-connection has its' own packet number space for transmitting 1-RTT packets.

ACK frame [I-D.ietf-quic-transport] MUST be returned via the same sub-connection on which the corresponding packets were sent.

MP_ACK frame can be returned via either a different sub-connection, or the same sub-connection, based on different strategies of sending MP_ACK frames.

Note: Only MP_ACK frame returned via the same sub-connection can be used to calculate RTT(round trip time).

7.3. Flow control

TODO

8. New frames

All the MP frames MUST be sent in 1-RTT packet, and MUST NOT use other encryption levels.

If an endpoint receives MP frames from packets of other encryption levels, it MAY return MP_PROTOCOL_VIOLATION as a connection error and close the connection.

8.1. MP_SUB_CONN_NEW frame

MP_SUB_CONN_NEW frame(type=0x2a) is used to establish a new sub-connection. The MP_SUB_CONN_NEW frame will specify a SCI and include a 64-bit random value.

MP_SUB_CONN_NEW frames are formatted as shown in Figure 5.

```
MP_SUB_CONN_NEW Frame {
  Type (i) = 0x2a,
  Sub_Connection_Index (i),
  Data (64),
}
```

Figure 5: MP_SUB_CONN_NEW Frame Format

8.2. MP_SUB_CONN_ACCEPT frame

MP_SUB_CONN_ACCEPT frame (type=0x2b) is used by endto accept a new sub-connection, as a response to MP_NEW_SUB_CONN frame.

MP_SUB_CONN_ACCEPT frames are formatted as shown in Figure 6, which is identical to the MP_NEW_SUB_CONN frame (Section 8.1).

```
MP_SUB_CONN_ACCEPT Frame {
  Type (i) = 0x2b,
  Sub_Connection_Index (i),
  Data (64),
}
```

Figure 6: MP_SUB_CONN_ACCEPT Frame Format

8.3. MP_SUB_CONN_CLOSE frame

MP_SUB_CONN_CLOSE frame(type=0x2c..0x2d) is used to close a sub-connection, which is formatted by adding a SCI field to QUIC-Transport [I-D.ietf-quic-transport] CONNECTION_CLOSE frame. The SCI is used to distinguish sub-connections, so each sub-connection can be closed independently.

MP_SUB_CONN_CLOSE frames are formatted as shown in Figure 7.

```
MP_SUB_CONN_CLOSE Frame {
  Type (i) = 0x2c..0x2d,
  Sub_Connection_Index (i),
  Error Code (i),
  [Frame Type (i)],
  Reason Phrase Length (i),
  Reason Phrase (..),
}
```

Figure 7: MP_SUB_CONN_CLOSE Frame Format

8.4. MP_ACK frame

MP_ACK frame allows for acknowledgements on different sub-connections.

MP_ACK frame is formatted by adding a SCI field and QoE signal fields to QUIC-Transport [I-D.ietf-quic-transport] ACK frame.

MP_ACK frames are formatted as shown in Figure 8.

```
MP_ACK Frame {
  Type (i) = 0x22..0x23..0x24..0x25,
  Sub_Connection_Index(i),
  Largest Acknowledged (i),
  ACK Delay (i),
  ACK Range Count (i),
  First ACK Range (i),
  ACK Range (..) ...,
  [ECN Counts (..)],
  [QoE Control Signals Length(8)],
  [QoE Control Signals (..)],
}
```

Figure 8: MP_ACK Frame Format

Type(i) = 0x22 , with no ECN Counts and no QoE Control Signals

Type(i) = 0x23 , with ECN Counts and no QoE Control Signals

Type(i) = 0x24 , with no ECN Counts and QoE Control Signals

Type(i) = 0x25 , with ECN Counts and QoE Control Signals

8.5. MP_ADD_ADDRESS frame

TODO

8.6. MP_REMOVE_ADDRESS frame

TODO

9. IANA Considerations

This document makes no request of IANA.

10. Informative References

[I-D.deconinck-quic-multipath]
De Coninck, Q. and O. Bonaventure, "Multipath Extensions for QUIC (MP-QUIC)", draft-deconinck-quic-multipath-05 (work in progress), August 2020.

[I-D.ietf-quic-load-balancers]
Duke, M. and N. Banks, "QUIC-LB: Generating Routable QUIC Connection IDs", draft-ietf-quic-load-balancers-04 (work in progress), August 2020.

[I-D.ietf-quic-transport]
Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport", draft-ietf-quic-transport-32 (work in progress), October 2020.

[MPQUIC-Scheduler]
Wang, J., Gao, Y., and C. Xu, "A Multipath QUIC Scheduler for Mobile HTTP/2", Proceedings of the 3rd Asia-Pacific Workshop on Networking 2019 (APNet '19). Association for Computing Machinery, New York, NY, USA, 43-49., 2019, <https://doi.org/10.1145/3343180.3343185>.

[MPTCP-LIA]
Raiciu, C., Handly, M., and D. Wischik, "Coupled Congestion Control for Multipath Transport Protocols", RFC 6365, October 2011, <https://tools.ietf.org/html/rfc6356>.

[MPTCP-OLIA]
Khalili, R., Gast, N., and J. Boudec, "Opportunistic Linked-Increases Congestion Control Algorithm for MPTCP", July 2014, <https://datatracker.ietf.org/doc/html/draft-khalili-mptcp-congestion-control-05>.

[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/info/rfc2119>.

[RFC8684] Ford, A., Raiciu, C., Handley, M., Bonaventure, O., and C. Paasch, "TCP Extensions for Multipath Operation with Multiple Addresses", RFC 8684, March 2020, <https://tools.ietf.org/html/rfc8684>.

Authors' Addresses

Qing An
Alibaba Inc.

Email: anqing.aq@alibaba-inc.com

Yanmei Liu
Alibaba Inc.

Email: miaoji.lym@alibaba-inc.com

Yunfei Ma
Alibaba Inc.

Email: yunfei.ma@alibaba-inc.com

Zhenyu Li
ICT-CAS

Email: zyli@ict.ac.cn