Increasing Throughput & Redundancy with Multipath QUIC

Hyeon Su Kim, Marcel Kempf*, Kilian Holzinger*
*Chair of Network Architectures and Services
School of Computation, Information and Technology, Technical University of Munich, Germany
Email: hs.kim@tum.de, kempfm@net.in.tum.de, holzinger@net.in.tum.de

***Abstract*—With the increasing amount of connected devices over the years, existing internet protocols have been changed and new protocols have emerged. One of the most recent protocols is the QUIC protocol. QUIC is a transport protocol designed to overcome the shortcomings of the more commonly used Transmission Control Protocol (TCP).**

Although QUICs functionalities serve as a solid foundation for a connection, dedicated working groups have wondered about the establishment of a Multipath QUIC connection. In this paper various versions of the IETF draft as well as practical implementations of Multipath QUIC will be analyzed. First the mechanisms that the current IETF draft specifies will be summarized and important keypoints during the development will be pointed out and analyzed. And lastly the MPQUIC extension will be evaluated based on potential use cases and the resulting benefits as well as concerns that arise.

***Index Terms*—QUIC, Multipath Transport Protocol, Multipath TCP**

1. Introduction

Although TCP has been the most commonly used reliable transport protocol for a long time, many points of improvement have been noticed. QUIC is a reliable transport protocol with additional security functionality that addresses these points and deploys different mechanisms to improve on these points [1, section 1].

Development on QUIC began in 2012 and was finally standardized in 2021 by the IETF [2]. The protocol offers a reliable connection between two hosts that claims to be faster and more secure than the TCP and TLS stack. While QUIC offers a reliable connection with one path, without modification it lacks the ability to utilize more than one interface concurrently. Therefore a dedicated group of researchers have been working on an extension that provides the ability to form a QUIC connection between two hosts with the concurrent use of multiple paths [3, section 1]. MPQUIC aims to utilize up to all network interfaces of a hosts machine in order to establish a multipath connection by mostly reusing the QUIC protocol and avoiding changes to the protocol. This concept has been a work in progress since 2017 and is heavily worked on by the dedicated group.

The anticipated potential for an even higher throughput of data and even higher resistence against network location changes awakes interest in this extension.

In this paper, we will observe the past and present drafts [3] in order to achieve an understanding of the operations that take place with the deployment of the extension and the problems and disagreements that occured during the development of the draft. Afterwards we will compare MPQUIC to other multipath transport protocols such as MPTCP [4]. The last section this paper evaluates the current MPQUIC draft by analyzing use cases in terms of the benefits and stating concerns about the extension in its current form.

This paper offers insight into similar works such as the archived MPQUIC IETF drafts [3] and the original paper proposing the MPQUIC draft [5].

2. Background

This section aims to establish background knowledge about MPQUICs multipath operations and additionally QUICs mechanisms, since these are vital in order to understand MPQUICs procedures.

2.1. QUIC

One of the interesting features of QUIC is connection migration. A devices' network location can change due to middleboxes assigning new network parameters or the device changing the physical location to another network. These circumstances can cause issues to TCP. Connections between machines are classified by the 4-tuple, consisting of the ip-address and the port number of both hosts. TCP can classify the managed connection only by these 4 values [6, Section 3.1] and a change in one of those will identify the previous connection as a new one. The connection will be discarded and eventually the disconnected host will have to reconnect to the service. This handling of the changed parameters is highly inefficient and QUIC provides a better way of handling such an event.

QUIC uses the Connection ID parameter in order to identify the connection and the 4-tuple to identify the path [1, Section 3.1]. In a situation where the 4-tuple changes, the protocol can identify the changed connection as a migrated one. After validating the new path with a simple challenge and response procedure the modified connection can be used again [1, Section 9]. This validation procedure is called Path Validation and it is mandatory for any migrated connection that needs to be used again [1, Section 9]. The packets that contain these frames are classified as probing packets.

QUIC defines a unique packet structure that consists of packet headers which carry a variable amount of frames. One packet consists of a UDP header that carries one or more headers. There are two types of

 doi: 10.2313/NET-2024-09-1_03

headers defined in QUIC, the short header and long header [1, Section 17]. Frames serve a single function or carry one parameter that are utilized by the transport protocol for certain functionalities. The most relevant frames in this paper are the ACK frame, the PATH_CHALLENGE, PATH_RESPONSE, and the RETIRE_CONNECTION_ID frame. The ACK frame acknowledges the last received frame, PATH_CHALLENGE and PATH_RESPONSE frames carry the data for the Path Validation procedure and RETIRE_CONNECTION_ID informs the peer that the sending host will no longer use the connection id that was agreed upon during the handshake [1, Section 19].

A unique feature that QUIC deploys is packet protection. With the assistance of TLS, packets are encrypted with Authenticated Encryption with Associated Data (AEAD) [7, Section 5]. Version Negotiation packets are the exception to this procedure and will not be encrypted.

QUIC also implements features that reduce latency [1, Section 7, 13]. Since these features are unaffected by the extension, it is sufficient to know solely about the existence of the features.

2.2. Multipath QUIC

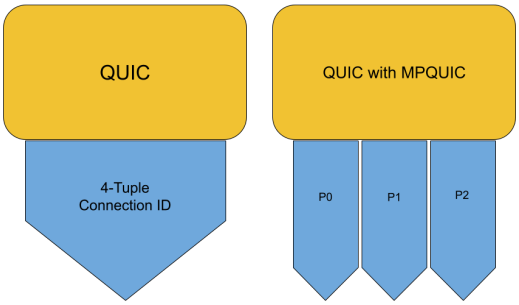

Figure 1: Structure of QUIC and MPQUIC

The Multipath extension aims to establish a connection between two hosts with multiple paths. Path in this context is the connection between one network interface to another [3, Section 1]. This is realized by instantiating paths that are different from each other in their respective 4-tuple, Destination Connection ID and packet number space [3, Section 4.4]. With more than one path being utilized, it requires operations such as RTT-measurement to be processed on each path individually.

2.3. Multipath Connection Establishment

The Multipath QUIC extension is only usable after successful negotiation during the handshake phase. Note that during this phase, the deployed transport protocol does not use any mechanisms of MPQUIC. The following conditions have to be met during the handshake phase for a successful negotiation:

- each of the hosts provide a set value for the transport parameter enable_multipath
- each of the hosts use a non-zero length connection id for source and destination
- negotiated cipher suites are at least 12 bytes long

If either of these conditions are not met, the extension can not be used [3, Section 3] and it is highly likely that a single path QUIC connection will be established.

Additionally during the handshake the parameter active_connection_id_limit dictates how many paths can be established.

2.4. Multipath Operations

Path instantiation is done by sending a non-probing packet to the server from unused path. The server then detects this as an attempt to create a new path and sends a packet with a path challenge in order to validate the new path. Only after successfully responding to the challenge the newly founded path can be used.

Each of the hosts can notify the peer on which path they would like to receive data from. Hosts can notify the paths preference with the frames PATH_AVAILABLE and PATH_STANDBY. The first frame signals that the specified path can be used to send data and the second frame signals that the peer would preferably not like to receive data on the path [3, Section 4.2].

It is also possible to close a path that no data transmission takes place on the specified path. The conventional way to close a path is to send a PATH_ABANDON frame. After receiving this frame, the host has to wait for three times the probe timeout interval for potential packets inflight that are related to the closed path. After receiving the acknowledgement for the PATH_ABANDON frame a RETIRE_CONNECTION_ID frame is sent to finally release the resources for the closed path [3, Section 4.3].

Other events that close a path are closure of the entire connection through CONNECTION_CLOSE frame or a stateless reset [3, Section 4.3].

A special case of closure is the sole usage of RETIRE_CONNECTION_ID frame which renders the specified connection id unusable. The associated path can't be used anymore, unless there is another connection id to be assigned. If this frame is used the sending host should consider inflight packets related to the retired connection id and a waiting period should be considered [3, Section 4.3.3].

Another reason for a path closure would be for an idle timeout. This is detected by the lack of non-probing packets received or lack of acknowledgements from sent packets [3, Section 4.3.4].

Each path has to be managed separately and operations such as path RTT-measurements and congestion control must be processed for each individually path. As previously mentioned, each path has its own packet number space that packets with number N can be observed on each path. This complicates the identification process of packets since packet numbers can not be used as a unique identifier anymore. Therefore MPQUIC uses additionally the Destination Connection ID in order to identify packets [3, Section 5].

In total, there are 4 different states that a path can be classified with. A path with the Validating state refers to a path that has been just instantiated by either a sent or received PATH_CHALLENGE frame. The path then transitions over to the Active state when a response has

been received. These paths can be used and preferences can be set to them just as mentioned before. A path then transitions to the Closing state when the host sends a PATH_ABANDON frame. And at last, according to the IETF draft [3, Section 4.4] there are 3 different events that lead to the Closed state:

- Path Validation process failure
- Timeout during the Active state
- Sending a RETIRE_CONNECTION_ID frame

An important note to MPQUIC that one must keep in mind is that the ACK_MP frame does not have to follow the same path that it acknowledges [3, Section 5.1].

2.5. Packet Number Spaces

The draft states that each path has their own packet number space. With the change from single to multiple Packet Number Spaces, some of the mechanisms that are dependent on the Packet Number are modified.

MPQUIC packets carry the Destination Connection ID in addition to the packet number for association to the correct path [3, Section 5]. With the addition of Destination Connection ID the ACK_MP frame has to include the Destination Connection ID sequence number in order to acknowledge packets for the specified path [3, Section 5.1].

The AEAD for the packet protection normally requires the packet number for the calculation of the nonce. This changes when using MPQUIC since packet numbers can occur multiple times per path meaning the nonce is not unique for the processed packet. Therefore AEAD mechanism under MPQUIC calculates the nonce additionally with the destination connection ID sequence number [3, Section 5.2].

2.6. Scheduling

The draft vaguely specifies the necessity of a scheduler with the usage of Multipath. The general scheduling procedure consists of manipulating congestion windows of the active paths and distributing the packets accordingly. The distributing logic of the scheduler does not affect control frames due to their urgency.

An important note is that the choice of scheduler algorithm depends on the application and potentially the role of the hosts as either the client or the server [3, section 7.4].

3. Development

In order to understand the design choices during development this section will cover aspects of MPQUIC that were especially difficult to handle during development of the several versions of the ietf drafts and the versions of QUIC implementing the extension. Work on an IETF draft document started in 2017 by Quentin De Coninck and Olivier Bonaventure. In 2020 two additional versions covering the Multipath QUIC extension were created. One authored by Christian Huitema and Mirja Kühlewind, the third draft document by Yanmei Liu and Yunfei Ma from the Alibaba coorperation. Those 3 drafts are different in the technicalities in how the multipath connection is created. In 2021 the three drafts were merged into one with ideas from each of the drafts influencing the final version.

All of the created drafts share some common ground:

- negotiation during handshake with the enabling parameter
- validation of a path before usage
- notifying preferences for data reception on certain paths
- a communication flow is not limited to one path
- consideration for a scheduler

Even though all three groups goal were equal, there were significant differences between the drafts. Finally in 2021 all three branches of drafts were merged into one with clear implementation instructions that compromise the ideas of each draft. During the development, two features of the extension brought up considerable amount of discussion: Packet Number Spaces and Path Identifier.

3.1. Single or Multiple Packet Number Spaces

During the development period the 3 branches of drafts each implemented different mechanisms:

- one single Packet Number Space for all paths
- separate Packet Number Spaces for each path
- option to use both Packet Number Spaces

The reason for experimentation with the two methodologies is due to the different advantages that these offer. In [8, Section 1] it is stated that the advantage of a unified Packet Number Space is the ability to hide the Connection ID and using a zero length Connection ID. Another advantage that Christian Huitema brought up was that technically the default QUIC was already aware of multiple paths with the connection migration feature and also used a single Packet Number Space for processing packets [9]. Therefore the implementation of multiple Packet Number Spaces would have seemed like a severe modification to the QUIC protocol.

Unfortunately acknowledgements with a shared Packet Number Space are more difficult to handle since the in-flight packets can not be expected to be received in order due to the different paths that the packets can travel through. Even with several countermeasure options stated in the draft [8, section 7.1.1], 4 implementation difficulties were found and stated in an Email by Yunfei Ma [10]:

1) Inaccuracy with RTT measurements
2) ACK range with holes of significant size
3) degrading speed with increasing data size
4) ACK size can be suppressed

With this technical report the support for multiple Packet Number Spaces increased significantly since the issues could not be ignored and further development on the extension was slowed.

Fortunately, separated Packet Number Spaces did not present with the same issue. Due to the addition of identifying parameters of the paths, the ACK_MP frames have to include the identifier in order to adopt the same acknowledgement procedure as the default QUIC protocol [3, Section 5.1]. Although the disadvantage of exposing

the connection ID and the path identifier is significant, the latest draft has adopted multiple Packet Number Spaces and offers no option to use a single one [3].

The adaption of either single or multiple Packet Number Spaces was heavily discussed among the QUIC working group and due to disagreements the decision was put to a poll [11] and due to overwhelming support for the removal of single packet number space, it was removed in 2023.

3.2. Path Identifier

During development the Path Identifier was created as a parameter identifying the path. In 2023 path identifiers were removed from the draft and implementations that implement the ietf MPQUIC draft. The reasons for the removal were the added complications when considering the usage of connection IDs and Path IDs [12]. More accurately the usage of path ID relies on the 4-tuple which is unreliable as an identifier in the current network structure. Additionally it is difficult to differenciate the events where the 4-tuple changes with a path ID [12]. Therefore the working group decided to use the Destination Connection ID sequence number for path identification.

However the use of Destination Connection ID sequence number seems to be a short-term solution. This parameter is not static for a long period of time and changes frequently due to e.g. Connection ID rotations. Moreover since the Packet Number Space is bound to the Connection ID, changes to the Connection ID would also affect the Packet Number Spaces. Therefore in [13] Marten Seemann proposes an explicit Path Identifier. The new Path ID is an independent parameter that is solely bound to the path that it is related to. This would also allow the calculation of Packet Number Spaces without the concern of involuntary changes. are unrelated to the paths directly. This explicit Path Identifier was eventually tested and presented on the 119 IETF meeting [14] and it is highly likely that the IETF draft will implement the new parameter in the future.

4. Evaluation

This section will cover the evaluation of MPQUIC. This includes comparison with MPTCP, existing implementations and use cases.

4.1. Comparison with MPTCP

Many similarities between the two extensions are easy to notice. Both extensions can not be used without a successful negotiation during their respective handshake phase. Additionally both extensions map the sent packets to the different paths in order to acknowledge packets according to the paths [4].

While both MPQUIC and Multipath TCP (MPTCP) establish multiple paths with their respective transport protocols, the structure is slightly different.

As mentioned in the previous section the MPQUIC extension enhances the default QUIC connection to be mapped over more than one path. This means that there is only one instance of a reliable transport protocol running at both hosts devices [3]. MPTCP manages multiple so called subflows that are on surface a whole TCP connection. Therefore the MPTCP extension is an additional protocol that governs over several TCP instances [4].

4.2. Implemetations

In total there are 3 open-source QUIC libraries that actively follow the IETF draft and implement the specified mechanisms: picoquic, quiche and xquic.

The MPQUIC extensions are implemented in each of these repositories.

All repositories implement the current version of the MPQUIC draft with the base functionalities such as modified handshake, initializing new paths, path management. It should be noted that xquic is the only version implementing various schedulers [15]. In fact xquic has the shortest development cycle from all three versions. This could be attributed to the funding that the repository receives from Alibaba inc [16].

4.3. Use Cases

With the availability of the Multipath QUIC extension various applications would make use of the extended transport protocol. Applications in need of high throughput will highly favor the protocol since servers could make use of its multiple network interfaces in order to deliver data at a faster rate. Especially services that benefit greatly from a consistently high data rate such as video streaming services are potential users of MPQUIC.

Another use case that can come to mind is to build dedicated communication flows. Since MPQUIC allows to notify about the preferences of paths in terms of reception of data, one can build an application that uses a dedicated path for receiving data. This means with dedicated control the dataflow of sending and receiving packets do rarely collide leading to a smoother operation.

Applications that are dependent on the constant availability of the service will benefit greatly from the use of MPQUIC. Due to the awareness of past open paths hosts can react faster to a change in network addresses. For example, an application on a smart phone would be able to detect a departure from the local wireless network. From this deduction the application could open a second path utilizing the LTE capability. Outside of the wireless networks range the application would still have an active connection and can maintain a data exchange without any delay.

4.4. Concerns

The main concern for MPQUIC would be its security. Moreover using MPQUIC could leave the protocol more vunerable to spoofing atempts. In the default QUIC protocol there are several countermeasures against spoofing atempts such as the limit on a newly migrated connection and ability to detect such an attempt [1, section 9.3]. In the current IETF draft [3] there is neither a way to detect a spoofing attempt nor is there a countermeasure to protect against such attacks. Especially concerning is the path initialization procedure that only requires a packet

to be sent [3, section 4.1]. This means that a malicious thirdparty could potentially open a new path and the server would have difficulties to distinguish between the legitimate client and the third party.

Another concern that should be mentioned is the increase of congestion through the use of MPQUIC. A simple scenario would be a popular video streaming service that uses the MPQUIC extension. While the service would utilize the increased throughput of the transport protocol, it would also mean that the immediate network connections would be more congested. Especially at times when the streaming service would see increased usage the network would experience a huge increase of data flow and would therefore be more congested.

5. Conclusion and future work

The Multipath QUIC extension realizes the aim for high throughput and redundancy successfully. Especially promising are the ongoing findings in the IETF MPQUIC drafts [3] and discussions in the IETF mailing list [17]. These ongoing improvements build confidence that concerns and issues will be addressed soon. With this speed of development one can imagine that the stanardization will take place in the near future. With the standardization many application developers that either are already using QUIC or are in search of a realiable Multipath Transport Protocol will benefit greatly from the extension and common users of services that require transportation of data will be able to experience better services going forward.

References

[1] J. Iyengar and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport," RFC 9000, May 2021. [Online]. Available: https://www.rfc-editor.org/info/rfc9000

[2] Wikipedia contributors, "Quic — Wikipedia, the free encyclopedia," 2024, [Online; accessed 28-March-2024]. [Online]. Available: https://en.wikipedia.org/w/index.php?title=QUIC&oldid=1215763552

[3] Y. Liu, Y. Ma, Q. D. Coninck, O. Bonaventure, C. Huitema, and M. Kühlewind, "Multipath Extension for QUIC," Internet Engineering Task Force, Internet-Draft draft-ietf-quic-multipath-06, Oct. 2023, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-quic-multipath/06/

[4] A. Ford, C. Raiciu, M. J. Handley, O. Bonaventure, and C. Paasch, "TCP Extensions for Multipath Operation with Multiple Addresses," RFC 8684, Mar. 2020. [Online]. Available: https://www.rfc-editor.org/info/rfc8684

[5] Q. De Coninck and O. Bonaventure, "Multipath quic: Design and evaluation," in *Proceedings of the 13th International Conference on Emerging Networking EXperiments and Technologies*, ser. CoNEXT '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 160–166. [Online]. Available: https://doi.org/10.1145/3143361.3143370

[6] W. Eddy, "Transmission Control Protocol (TCP)," RFC 9293, Aug. 2022. [Online]. Available: https://www.rfc-editor.org/info/rfc9293

[7] M. Thomson and S. Turner, "Using TLS to Secure QUIC," RFC 9001, May 2021. [Online]. Available: https://www.rfc-editor.org/info/rfc9001

[8] Y. Liu, Y. Ma, Q. D. Coninck, O. Bonaventure, C. Huitema, and M. Kühlewind, "Multipath Extension for QUIC," Internet Engineering Task Force, Internet-Draft draft-lmbdhk-quic-multipath-00, Oct. 2021, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/draft-lmbdhk-quic-multipath/00/

[9] QUIC IETF Mailinglist. (2020) Re: Preparing for discussion on what to do about the multipath extension milestone. [Online]. Available: https://mailarchive.ietf.org/arch/msg/quic/licsYSzeigfuLxnizDDv-r0dlVE/

[10] ——. (2022) Alibaba's Technical Report of single packet number space implementation (SPNS) for multi-path QUIC. [Online]. Available: https://mailarchive.ietf.org/arch/msg/quic/LiEYJ2k8ldbz-SWeiXygeRAGHdk/

[11] ——. (2023) Consensus call: Removing 0-length CIDs (Single Packet Number Space) From MPQUIC. [Online]. Available: https://mailarchive.ietf.org/arch/msg/quic/_Dsr-cwm055bRzRKU-hxsVWjYqI/

[12] IETF MPQUIC Working Group. (2023) draft-ietf-quic-multipath-04 presentation slides. [Online]. Available: https://datatracker.ietf.org/meeting/116/materials/slides-116-quic-multipath-quic-00

[13] ——. (2023) separate Path IDs from Connection IDs. [Online]. Available: https://github.com/quicwg/multipath/issues/214

[14] ——. (2024) Draft-ietf-quic-multipath-ietf199. [Online]. Available: https://datatracker.ietf.org/meeting/119/materials/slides-119-quic-multipath-quic-00

[15] Alibaba inc. (2023) xquic schedulers. [Online]. Available: https://github.com/alibaba/xquic/tree/main/src/transport/scheduler

[16] ——. (2023) xquic schedulers. [Online]. Available: https://github.com/alibaba/xquic

[17] IETF MPQUIC Working Group. (2023) MPQUIC Mailinglist. [Online]. Available: https://mailarchive.ietf.org/arch/browse/quic/?q=multipath