Workgroup: Internet Engineering Task Force
Internet-Draft:
draft-kuhn-quic-bdpframe-extension-05
Published: 4 March 2024
Intended Status: Standards Track
Expires: 5 September 2024
Authors: N. Kuhn
Thales Alenia Space
E. Stephan
Orange
G. Fairhurst
University of Aberdeen
R. Secchi
University of Aberdeen
C. Huitema
Private Octopus Inc.

# Signalling CC Parameters for Careful Resume using QUIC

## Abstract

This document describes an extension for QUIC. This enables the exchange of Congestion Control (CC) parameters related to the characteristics of the between the sender and the receiver during a connection. These CC parameters allow a receiver to prepare to use additional capacity that is made available when using Careful Resume. It can also allow a receiver to request that previously computed CC parameters, are not used when the receiver has additional information about the current path or traffic that is to be sent.

## 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 5 September 2024.

**Table of Contents**



## 1. Introduction

This document extends the Careful Resume method [I-D.ietf-tsvwg-careful-resume] to allow sender-generated Congestion Control parameters (CC params) to be stored at the receiver, and used by the sender to resume a subsequent connection.

Transferring these CC params to a receiver can release the sender from needing to retain this state for each receiver. The method also allows a receiver to implement a policy that informs a sender whether the receiver desires the sender to reuse any previously saved CC params. A sender can independently decide whether to use Careful Resume.

This specifies both sender and receiver changes. If a sender does not support the current method, it will ignore the requests made by the receiver. If a receiver does not support the method, the sender does not receive any of the specified requests.

### 1.1. Three approaches

This section identifies three approaches to implement the storage of CC params:

(1) The saved CC params are stored at the sender ("Local storage"). They are not sent to a receiver, this does not use the method in this specification;

(2) The saved CC params are transmitted to the receiver in an opaque record. A receiver can choose to use the CC params when reconnecting, but is unable to read the value of the CC params;

(3) The saved CC params are transmitted to a receiver, in a format that allows parameters to be read. A receiver can choose to use CC params when reconnecting, and is able to read the value of the CC params to decide whether to accept or not the use of these parameters. This is the method described in this specification.

### 1.2. Example Cases where a Receiver might decide not to request use of saved CC Params

A receiver using approaches 2 or 3 can choose whether to request or avoid use of the saved CC params. This can be based on local knowledge of the network conditions, connectivity, or connection requirements.

Four cases are identified where Careful Resume would not be appropriate and using the saved CC params could increase congestion:

1. The network path has changed and the new path is different. This might be evident from a change of local interface, a change in the sender IP address, or similar indication from the network. The saved CC params are not appropriate to the new path.

2. Measured data indicates a change in the network path characteristics, but the path has not changed. This case might be indicated by a change in the RTT, or evident by loss observed at the start of the new connection. (This case could also be detected in the Careful Resume Reconnaissance Phase.) The saved CC params are no longer appropriate to use.

3. There is a notification that the path characteristics have changed and it is expected that the current capacity has reduced. Examples include: notification of changes in the link propagation conditions, notification of a change in the operational mode of a link, or awareness of a new limitation in the hardware platform.

4. The saved CC params are not within the stated LifeTime. It is no longer reasonable to expect the path to have same characteristics.

In all these examples, use of the saved CC params could increase congestion, and a receiver could wish to reject the use of the saved CC params. The sender reverts to the normal QUIC CC behavior [RFC9002].

### 1.3. Example Cases where a Receiver might tune based on saved CC Params

Where a receiver is aware it is using a path with a high Bandwidth-Delay Product (BDP), approach 3 allows it to adapt other protocol parameters to better utilize the available capacity, e.g., to estimate a larger size for the flow credit.

Some designs of application do not use long-lasting transport connections. Instead, they use a series of shorter connections, typically each using the same path. This style of application can benefit when the receiver provides an estimate of the expected characteristics (e.g., to adapt the content of quality for a video application; or anticipate the time taken to complete the transmission of an object). An example scenario considers a client using Dynamic Adaptive Streaming over HTTPS (DASH) that is unable to receive sufficient data to reach the desired video playback quality supported by the path, because the video transport needs to independently determine the path capacity for each video chunk. In this example, a lower transfer rate is safe, but could lead to an overly conservative requested rate when the rate is based on the video transport performance. Using the CC param information, the client requests could be adapted based on the previously observed path characteristics, enabling a client to increase the requested quality of video chunks or to fill receiver buffers and avoid stalling playback.

Even if the capacity on the forward and return paths might be significantly different, clients experiencing a high BDP for their forward path would typically also experience a high BDP for their return path. The CC params related to the forward path could then potentially be used to initially adjust the transmission of data using the return path.

## 2. Notation and Terms

*BDP: Bandwidth Delay Product of the path (maximum path capacity);

*saved_cwnd: The capacity preserved from a previous connection, measured in bytes per RTT;

*saved_rtt: The preserved minimum RTT, corresponding to the minimum of a set RTT of measurements taken at the time when the saved_cwnd was estimated;

*LifeTime: The time at which CC params are no longer to be used;

*endpoint_token: An Endpoint Token for a receiver;

*secured hash: hash generated by the sender covering the list of CC params. The sender uses a private key to protect this hash.

### 2.1. Requirements Language

The keywords "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.

Variable-length integer encoding is defined in section 16 of [RFC9000].

## 3. Signalling CC Parameters

{XXX Editor Note: This presents the current transport method to signal use and to send CC params. There are various alternatives and the final choice is to be confirmed.}

The following subsections define four signals that carry transport control information. The transmission of these signals is not limited by flow control limits of the underlying QUIC transport.

### 3.1. Extension Activation (enable_careful_resume_indication)

A receiver indicates its willingness to accept the transmission of careful_resume_indications from the sender by sending a

enable_careful_resume_indication (0xTBD). This is a transport parameter sent in the 1 RTT connection.

- *0: Default value. By default, a receiver does not activate, the transmission of indications.

- *1: The value of 1 requests transmission of careful_resume_indications. When the receiver activates the extension, it agrees to receive careful_resume_indications carrying CC params.

- *All values larger than 1 are reserved for future use, and the receipt of these values MUST be treated as a connection error of type TRANSPORT_PARAMETER_ERROR [RFC9000].

The encoding for this Transport Parameter is described in Section 18 of [RFC9000].

This indication has no action when the sender does not support this specification.

### 3.2. The CC Params

```
CC_params {
  Lifetime (i),
  Saved Capacity (i),
  Saved RTT (i),
  Saved Endpoint Token (...)
}
```

Figure 1: Format of the CC Params

The CC params comprise the following fields:

- *LifeTime (lifetime): The extension_lifetime is a timestamp in milliseconds, encoded as a variable-length integer. This represents the validity in time of this extension.

- *Saved CWND (saved_cwnd): The saved_cwnd is a value in bytes per RTT, encoded as a variable-length integer. A sender bases this on the estimated available capacity for a connection.

- *Saved RTT (saved_rtt): The saved_rtt is a value in milliseconds, encoded as a variable-length integer. The saved_rtt is set to the min_rtt for a connection.

- *Saved Endpoint Token (saved_endpoint_token) : The Endpoint Token is defined in [I-D.ietf-tsvwg-careful-resume], and is discussed in a later section.

Careful use of the CC params is discussed in [I-D.ietf-tsvwg-careful-resume].

### 3.3. Transporting the CC Params (careful_resume_indication)

This section describes the careful_resume_indication. Once the extension has been activated, a sender in the Careful Resume Observe Phase MAY send a careful_resume_indication to the receiver including CC params. A sender MAY update the CC params by sending an additional careful_resume_indication within a connection. The rate of update SHOULD be limited (e.g., much less frequent than once for several RTTs).

The format of a careful_resume_indication is specified in Figure 2, and the format for the CC params is specified in Section 3.2.

```
BDP_FRAME {
  Type (i) = 0xXXX,
  Hash (...),
  CC_params,
}
```

Figure 2: Format of a Careful Resume Indication

*Hash (secured_hash): The sender constructs the secured_hash over the set of CC params. A sender uses this hash to detect whether the received CC params were modified. A sender MUST verify the secured_hash for the CC params, and MUST NOT use the CC params when it cannot verify the secured_hash.

Note: Section 19.21 of [RFC9000] states "An extension to QUIC that wishes to use a new type of frame MUST first ensure that a peer is able to understand the frame. An endpoint can use a transport parameter to signal its willingness to receive extension frame types. One transport parameter can indicate support for one or more extension frame types". Implicitly, it is a protocol violation error to use an extension frame type that has not been approved by the peer by receiving a enable_careful_resume_indication.

### 3.4. Request to use the saved CC Params (careful_resume_request)

This section describes the careful_resume_request.

A receiver MAY send a careful_resume_request to the sender to request use of saved CC params for the same pair of endpoints when the server is expected to be in the Careful Resume Reconnaissance Phase. The sender decides whether to use or not the received CC params.

Note: A receiver ought to treat the transmission of a careful_resume_request as an indication that the path may have a higher BDP, and therefore might accordingly allocate additional flow credit to prevent potential use of the Careful Resume method becoming limited by flow control.

The format of a careful_resume_request is shown in Figure 2, and the format for the CC params is shown in Figure 1.

### 3.5. Request to avoid using the saved CC Params (careful_resume_avoid)

This section describes a transport parameter to a sender requesting it avoids using the saved cc params. No parameters are exchanged. This request is sent in the 1-RTT connection.

Note: The sender determines whether to use or not Careful Resume. The signal has no action when a sender does not support this specification.

## 4. Discussion

### 4.1. Use of the saved CC Params

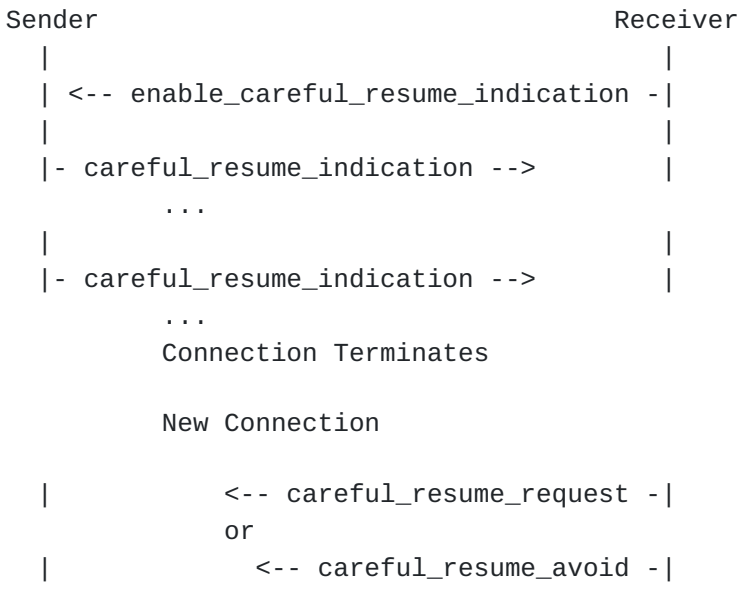


Figure 3: Sequence Diagram

1. Recption of a enable_careful_resume_indication, permits a sender to start sending CC params using a careful_resume_indication. This is done in the Observe Phase of the Careful Resume method. Each indication includes the current RTT, measured capacity, requested lifetime, and receiver Endpoint Token are computed and are respectively stored as the

saved_rtt, saved_cwnd, LifeTime and saved_endpoint_token. These form a set of CC params. The sender also computes a secured hash over these CC params and includes this within the careful_resume_indication.

2. When transmitted, the careful_resume_indication is encrypted by QUIC transport using TLS [RFC8446] [RFC9001]. The secured_hash is used by a sender to verify that the returned CC params were not modified by the receiver.

3. A receiver is unable to verify the secured_hash and is not permitted to modify any CC params, but it is expected to store these params and their hash. Access to this information would normally be indexed on the receiver's view of the path to the server. It can read any non-encrypted CC params (see Section 1.3).

4. When the receiver makes a subsequent connection, it can send the CC params (and the secured_hash) using a careful_resume_request to the sender at the start of a new connection. These CC params need to be received during the Reconnaissance Phase of the Careful Resume method.

5. Upon reception, a sender MUST verify the secured_hash, and only use the CC params when this is valid. The Careful Resume method specifies the rules for utilizing verified CC params.

6. The receiver is permitted to utlize local information and then decide to send a careful_resume_request or a careful_resume_avoid. The latter requests the sender does not use Careful Resume. A receiver could alternatively decide to not send a careful_resume_request expressing no preference.

### 4.2. Identifying the Path

This extension is designed to avoid an opportunity for the current connection to be a vector for an amplification attack. The QUIC address validation process, used to prevent amplification attacks, SHOULD be performed [RFC9000].

An example implementation where the sender computes an Endpoint Token to uniquely identify the receiver is provided in Section 4.3.

In a simple network scenario, the sending endpoint could use the IP source address to identify a path. This could work when one globally-allocated IP address is set per interface. There are many cases where the IP address would not an acceptable to identify a path. Section 8 of [RFC9040] describes cases where the IP address is not a suitable value when performing TCP control block sharing. In general the IP address of the sender is made public in the network-

layer header of IP packets. When sharing internal state, [RFC6973] identifies relevant privacy considerations.

Examples of network uses where a source address is not a suitable endpoint token include:

* The sending endpoint might not be remotely identifiable from its IP address because a device on the network path translates the address using a form of NAT/NAPT. In this case, a private IP address might be used, which does not identify a specific endpoint.

* In some cases, a sender can choose to vary the source address over time to avoid linkability in the observable IP header, e.g., when the source address embeds private information, such as an endpoint's MAC address/EIDID.

Note: There are use-cases where for the purpose of identifying a path, the token does not need to be globally unique, but needs to be sufficiently unique to prevent attempts to misrepresent the path being used such as an attack on the congestion controller. Using a smaller size of token can add to the ambiguity set, reducing this likability.

Note: A different Endpoint Token is used for each direction of transmission. A receiver could decide not to uses this method to avoid exposing additional link-able information (but also preventing use of any mechanism that relies on the token).

### 4.3. Example use of an Endpoint Token

The sender computes an Endpoint Token that seeks to uniquely identify the path that it uses to communicate with the receiver (1) this is associated with the path information it sends. The Endpoint Token ought to be encrypted to avoid sending link-able information observable to eavesdroppers on the path. The receiver stores the path information together with the Endpoint Token, together with the sender's address/name (2). When the receiver later wishes the sender to use this, it returns the information to the sender (3) together with the Endpoint Token. The sender recomputes the Endpoint Token and compares this with the received Endpoint Token before using the CC params.

1. The Sender transmits the Endpoint Token to the Receiver;

2. The Receiver holds an Endpoint Token;

3. The Receiver transmits the Endpoint Token to the Sender.

### 4.4. Security Topics Related to use of the Endpoint Token

A number of security-related topics have been identified concerning the potential exposure of the identity on the path. This information can also be visible in the IP source address or higher-layer data, but can be hidden from a remote endpoint using methods such as MASQUE proxy. When used to inform the transport system using a layered proxy, the transport endpoint token refers to the endpoints of the outer QUIC header, and hence the proxy itself, not the end-to-end communication relayed by the proxy.

A sender or receiver might decide to not use this method if it has a strong requirement to prevent flows being linkable with previous flows to the same endpoint. A decision not to provide an Endpoint Token necessarily prevents the sender from requesting the receiver to return path information to allow the same CC parameters to be re-used, potentially strengthening privacy, but consequently eliminating any performance benefits.

### 4.5. Using the CC Params with Care

Care is needed in the use of any temporal information to assure safe use of the Internet and to be robust to changes in traffic patterns, network routing and link/node failures. There are cases where using the CC parameters of a previous connection are not appropriate, and a need to evaluate the potential for malicious use of this method. The specification for the QUIC transport protocol [RFC9000] therefore notes "Generally, implementations are advised to be cautious when using previous values on a new path".

## 5. Acknowledgments

The authors thank Gabriel Montenegro, Patrick McManus, Ian Swett, Igor Lubashev, Robin Marx, Roland Bless, Franklin Simo, Kazuho Oku, Q Misell, and Marten Seeman for their fruitful comments on earlier versions of this document.

The authors particularly thank Tom Jones for co-authoring previous versions of this document.

## 6. IANA Considerations

{XXX-Editor note: Text is required to register the four signals. Parameters are registered using the procedure defined in [RFC9000].}

## 7. Security Considerations

Security considerations for the CC method are discussed in the Security Considerations section of Careful Resume.

### 7.1. Protection from Malicious Receivers

The sender is required to check the integrity of the CC params using the hash computed over the block of CC params. Whilst data in transit is protected by TLS, this has is intended to protect the data at rest at the receiver. No specific hash algorithm is specified, because the value is only computed at the sender, and a sender can choose any suitable algorithm to meet its own security objectives.

## 8. References

### 8.1. Normative References


**[I-D.ietf-tsvwg-careful-resume]** Kuhn, N., Emile, S., Fairhurst, G., Secchi, R., and C. Huitema, "Convergence of Congestion Control from Retained State", Work in Progress, Internet-Draft, draft-ietf-tsvwg-careful-resume-07, 16 February 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-tsvwg-careful-resume-07>.

**[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>.

**[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/info/rfc8174>.

**[RFC9000]** 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/info/rfc9000>.


### 8.2. Informative References


**[RFC6973]** Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, DOI 10.17487/RFC6973, July 2013, <https://www.rfc-editor.org/info/rfc6973>.

**[RFC8446]** Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

**[RFC9001]** 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/info/rfc9001>.

**[RFC9002]** 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/info/rfc9002>.

**[RFC9040]** Touch, J., Welzl, M., and S. Islam, "TCP Control Block Interdependence", RFC 9040, DOI 10.17487/RFC9040, July 2021, <https://www.rfc-editor.org/info/rfc9040>.

## Appendix A. Change Log

This section to be removed prior to publication.

-00 Introduced session tickets and BDP_FRAMEs

-01 Reviewed receiver actions when a receiver holds CC parameters

-02 Interim version to align with terminology in Careful Resume

-03 Rewritten to align with Careful Resume and use the BDP_FRAME method. Removed annexe 1, and discussion of session tickets, preferring BDP_FRAMEs.

-04 (1) To align with Careful Resume to use saved_cwnd, after review of CR from Neil Cardwell. (2) To fix the alternative of using resumption tokens as proposed by Marten Seeman and Kazuho Oku. (3) To detail the client point of view and associated use-cases. (4) Rewritten to clarify story and separately identify: format; transport; use; discussion. This rev. would also allow a change of transport method if the WG advises this. (5) Specify two frames and two actions sending cc params.

-05 Editorial corrections.

## Authors' Addresses

Nicolas Kuhn
Thales Alenia Space

Email: nicolas.kuhn.ietf@gmail.com

Emile Stephan
Orange

Email: emile.stephan@orange.com

Godred Fairhurst
University of Aberdeen
Department of Engineering
Fraser Noble Building

Aberdeen
AB24 3UE
United Kingdom

Email: gorry@erg.abdn.ac.uk

Raffaello Secchi
University of Aberdeen
Department of Engineering
Fraser Noble Building
Aberdeen
AB24 3UE
United Kingdom

Email: r.secchi@erg.abdn.ac.uk

Christian Huitema
Private Octopus Inc.

Email: huitema@huitema.net