Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: November 22, 2019

N. Kuhn, Ed.
CNES
E. Stephan, Ed.
Orange
G. Fairhurst, Ed.
University of Aberdeen

May 21, 2019

Transport parameters for 0-RTT connections
draft-kuhn-quic-0rtt-bdp-02

Abstract

The NewSessionTicket record carries a field that tells a client the volume of early data that it can include in the 0-RTT messages when reconnecting to the same peer. There are cases where additional information can significantly improve the time-to-service. This memo discusses a solution where path adaptation parameters are also shared between the peers. There are use cases where this can accelerate the throughput of subsequent 0-RTT connections in both direction.

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 November 22, 2019.

Copyright Notice

Table of Contents

1. Introduction

The 0-RTT mechanism is designed to accelerate the throughput when establishing a connection. There are cases where 0-RTT alone does not improve the time-to-service, and additional information can therefore be beneficial.

Some network paths result in a reduced time-to-service because the default parameters controlling the initialization of the transport and congestion control are not suitable for the path characteristics. QUIC's congestion control is based on TCP NewReno [I-D.ietf-quic-recovery] and the recommended initial window is defined by [RFC6928]. A path with a large bandwidth delay product can therefore significantly increase the time-to-service (e.g. a path using satellite communication [IJSCN19] could observe a much longer page load time for complex pages).

This memo describes a solution where:

1. the server learns a fundamental characteristic of the path during the 1-RTT phase;

2. the server sends this information to the client at the end of the 1-RTT phase;

3. the server and the client exploit the information to improve the time-to-service during subsequent 0-RTT connections.

2. QUIC connection establishment: differences between 1-RTT and 0-RTT connections

This section recalls how 1-RTT and 0-RTT operate in QUIC [I-D.ietf-quic-transport].

QUIC leverages the 2 handshakes of TLS1.3 [I-D.ietf-quic-tls]: The 1-RTT handshake initiates a first set of credentials. When a handshake achieves successfully, the server pushes the learned information about the session to the client in an opaque session ticket (see section 4.6.1 of [RFC8446]). This information within the opaque ticket is meaningless to the client. A client willing to establish a fast re-opening of the session pushes back this opaque 'ticket' in a 0-RTT handshake and sends early application data.

In practice, the server sends the 'ticket' in a NewSessionTicket record [I-D.ietf-quic-tls]. The structure of the NewSessionTicket includes the opaque 'ticket' and an 'extensions' field. The NewSessionTicket carries an additional field named 'early_data' that indicates to the client the maximum size of application data to insert in the 0-RTT message.

3. End-to-end solution

This section proposes an improvement of the initialization of 0-RTT connections over high BDP networks where the client recalls, among other parameters, the BDP previously measured by the server during the 1-RTT handshake. The approach follows the tuning of the initial window described in [I-D.irtf-iccrg-sallantin-initial-spreading] that has been shown to improve performance both for high BDP and more common BDP [CONEXT15] [ICC16].

3.1. Description of the extension in the NewSessionTicket

This document defines an extension named "BDP_metadata" for the NewSessionTicket. This structure contains the following parameters: BDP, MTU, RTT, loss-rate.

3.2. Usage of the extension in the NewSessionTicket

At the end of a 1-RTT connection, a server can send information in a NewSessionTicket that describes the path to the client. The message includes an additional 'extensions' field named 'BDP_metadata'. The client stores this session ticket and the 'BDP_metadata' field.

When the client reconnects to the same server in 0-RTT mode, it pushes back this session ticket in the ClientHello and prepares itself to receive data in the context given by the 'BDP_metadata' field. (The client does not send the 'BDP_metadata' field back to the server). The server receives the session ticket and extracts the BDP context. As example, it can use this message to provide information that may allow the congestion controller to provide a throughput closer to the capacity of the path.

Because the path characteristics can change over time, and may hence become invalid for use in a subsequent connection, the server sets the age of the ticket (see section 4.2.11.1 of [RFC8446]) to a short duration. A server could also update the ticket when the path characteristics of connection are known to have changed.

4. Best current practice

This section provides examples of data that could be added in the opaque ticket field by the server. The details added by the server in the session ticket do not need to be standardized for interoperability between QUIC clients and servers because this information is opaque to the client. The presence of the "BDP_metadata" extension field in the NewSessionTicket informs the client that the server can actively take action to improve its throughput when the session will restart.

The following list describes information elements set by the server in the session ticket to accompany the signaling of field. These examples are illustrated in Figure 1 and their purpose is detailed in this section.

- A client aware of a high BDP path: Section 7.3.1 of [I-D.ietf-quic-transport] indicates that the "A client that attempts to send 0-RTT data MUST remember the transport parameters used by the server". In addition to the default transport parameters used by the server, a server that knows that the path has a large BDP can let the client adapt its parameters.

- PMTU: Knowledge of the PMTU of a previous path improves the time to service because it reduces the duration of the path validation process described in section 8.2 of [I-D.ietf-quic-transport].

- Connection RTT: The knowledge of the characteristics of a previous connection RTT can improve the throughput because a server can safely improve the slow start: e.g. using the pacing models of [I-D.irtf-iccrg-sallantin-initial-spreading] can result in high IW for high RTT paths and a common IW for paths with smaller RTT. The results presented in [ICC16] show that for both files of 15 KB and 750 KB, the proposed solution reduces the time to download by approximatively 2 seconds whether the RTT is 50ms or 500ms.

- Ticket_lifetime: The server sets a shorter validity duration to avoid receiving obsolete path characteristics; (e.g., this could reduce the validity to one day).

Figure 1: Example of opaque ticket content

5. What happens when BDP is used incorrectly?

This section discusses the impact of a server activating the 'BDP_metadata' field when the network underneath actually has a small BDP. This could happen when the server BDP estimate was incorrect, client has multiple paths to choose from and uses the ticket on a different path to which it was requested, or when the path characteristics change significantly.

Depending on how the extension is exploited, this could result in assigning additional resources (e.g. buffer space) that later is not

used. This could also motivate the requirement to pace the initial window to avoid transmitting data at a too high a rate.

6. Discussion

This mechanism follows the approach of the extension field 'early_data' of the NewSessionTicket of TLS1.3. While 'early_data' improves the egress traffic of a client, the 'BDP_metadata' proposal aims at improving ingress traffic to the client. Improving the ingress traffic can result in significant improvement to the quality-of-experience. For example, this enables the use of transport parameters, such as the RTT, PMTU and BDP to adapt the initial data transmission of a 0-RTT connection. In some large BDP deployment scenarios, this can halve the page load time of a web page download.

7. Acknowledgements

None.

8. IANA Considerations

TBD: text is required to register the extension BDP_metadata field.

9. Security Considerations

The security is provided by the 1-RTT phase. The measure of BDP is made during a previous connection. The exchange and the information are protected both by the TLS encryption and the NewSessionTicket (see section 4.6.1 of [RFC8446]).

The BDP information the server will received is protected in the opaque session ticket. The 'BDP_metadata' field is visible by the client only. An client that does not trust the server transport adaptation ignores any session ticket associated to a 'BDP_metadata' field.

The server does not have to honour all the received requests (e.g. when it is resource-limited).

10. References

10.1. Normative References

[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>.

10.2. Informative References

[CONEXT15]
Li, Q., Dong, M., and P. Godfrey, "Halfback: Running Short Flows Quickly and Safely", ACM CoNEXT , 2015.

[I-D.ietf-quic-recovery]
Iyengar, J. and I. Swett, "QUIC Loss Detection and Congestion Control", draft-ietf-quic-recovery-20 (work in progress), April 2019.

[I-D.ietf-quic-tls]
Thomson, M. and S. Turner, "Using TLS to Secure QUIC", draft-ietf-quic-tls-20 (work in progress), April 2019.

[I-D.ietf-quic-transport]
Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport", draft-ietf-quic-transport-20 (work in progress), April 2019.

[I-D.ietf-tls-ticketrequests]
Pauly, T., Schinazi, D., and C. Wood, "TLS Ticket Requests", draft-ietf-tls-ticketrequests-00 (work in progress), January 2019.

[I-D.irtf-iccrg-sallantin-initial-spreading]
Sallantin, R., Baudoin, C., Arnal, F., Dubois, E., Chaput, E., and A. Beylot, "Safe increase of the TCP's Initial Window Using Initial Spreading", draft-irtf-iccrg-sallantin-initial-spreading-00 (work in progress), January 2014.

[ICC16] Sallantin, R., Baudoin, C., Chaput, E., Arnal, F., Dubois, E., and A-L. Beylot, "Reducing web latency through TCP IW: Be smart", IEEE ICC , 2016.

[ICCRG100]
Kuhn, N., "MPTCP and BBR performance over Internet satellite paths", IETF ICCRG 100, 2017.

[IJSCN19] Thomas, L., Dubois, E., Kuhn, N., and E. Lochin, "Google QUIC performance over a public SATCOM access", International Journal of Satellite Communications and Networking , 2019.

[NCT13] Pirovano, A. and F. Garcia, "A new survey on improving TCP performances over geostationary satellite link", Network and Communication Technologies , 2013.

[RFC2488] Allman, M., Glover, D., and L. Sanchez, "Enhancing TCP Over Satellite Channels using Standard Mechanisms", BCP 28, RFC 2488, DOI 10.17487/RFC2488, January 1999, <https://www.rfc-editor.org/info/rfc2488>.

[RFC3135] Border, J., Kojo, M., Griner, J., Montenegro, G., and Z. Shelby, "Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations", RFC 3135, DOI 10.17487/RFC3135, June 2001, <https://www.rfc-editor.org/info/rfc3135>.

[RFC6349] Constantine, B., Forget, G., Geib, R., and R. Schrage, "Framework for TCP Throughput Testing", RFC 6349, DOI 10.17487/RFC6349, August 2011, <https://www.rfc-editor.org/info/rfc6349>.

[RFC6928] Chu, J., Dukkipati, N., Cheng, Y., and M. Mathis, "Increasing TCP's Initial Window", RFC 6928, DOI 10.17487/RFC6928, April 2013, <https://www.rfc-editor.org/info/rfc6928>.

[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>.

Authors' Addresses

Nicolas Kuhn (editor)
CNES

Email: nicolas.kuhn@cnes.fr

Emile Stephan (editor)
Orange

Email: emile.stephan@orange.com

Gorry Fairhurst (editor)
University of Aberdeen

Email: gorry@erg.abdn.ac.uk