Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: September 12, 2019

N. Kuhn, Ed.
CNES
E. Stephan, Ed.
Orange
March 11, 2019

Transport parameters for 0-RTT connections
draft-kuhn-quic-0rtt-bdp-01

Abstract

0-RTT is designed to accelerate the egress throughput at the establishment of a connection. There are cases where 0-RTT alone does not improve the time-to-service.

This memo discusses a solution where a fundamental characteristic of the path is learned during the 1-RTT phase and shared with the 0-RTT phase to accelerate the initial throughput during subsequent 0-RTT connections.

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 September 12, 2019.

Table of Contents

1. Introduction

0-RTT is designed to accelerate the throughput at the establishment of a connection. There are cases where 0-RTT alone does not improve the time-to-service.

As shown in [IJSCN19], the usage of a congestion control and transport initialization not adapted to satellite communication results in higher page loading time for heavy pages in a SATCOM context. QUIC's congestion control is based on TCP NewReno [I-D.ietf-quic-recovery] and the recommended initial window is defined by [RFC6928]. This may not be suitable for good quality of experience for users in high Bandwidth Delay-Product (BDP) networks.

This memo discusses a solution where a fundamental characteristic of the path is learned during the 1-RTT phase and shared with the 0-RTT phase to accelerate the initial throughput during subsequent 0-RTT connections.

2. QUIC connection establishment

This section recalls how 1-RTT and 0-RTT work.

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 information learned about the session to the client in an opaque session ticket (see section 4.6.1 of [RFC8446]). The pieces of information of the ticket are 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' which indicates to the client the maximal size of application data to insert in the 0-RTT message.

3. Large BDP connections

GEO-satellite based systems characteristics differ from terrestrial networks with:

- A large propagation delay of at least 250ms one-way delay;
- A high bit-rate in case of mobile users or when a user connects behind a box using Wi-Fi;
- Highly asymmetric links.

These characteristics have an impact on end-to-end congestion controls:

- Transport initialization: the 3-way handshake takes a long time reducing the time at which actual data can be transmitted;
- Maximum windows sizing: to fully exploit the bottleneck capacity, the high BDP may induce an important number of in-flights packets;
- Reliability: packet losses detection and correction is slow and the time needed for the end server to react to a congestion event may not be relevant;
- Getting up to speed: the exponential increase of the data rate transmission for a channel capacity probing is slowed down when the RTT is high.

4. TCP split solution

High BDP networks commonly break the TCP end-to-end paradigm to adapt the transport protocol. Splitting TCP allows adaptations to this specific use-case and assessing the issues discussed in section Section 3. PEP [RFC3135] are commonly deployed in SATCOM infrastructure for that purpose and their deployment can result in 50% page load time reduction in a SATCOM use-case [ICCRG100].

[NCT13] and [RFC3135] describe the main functionalities of SATCOM TCP split solutions. Shortly, for traffic going from a gateway to an end user behind a terminal, the TCP split intercepts TCP SYN to act as the end user and adapt the data rate transmission to the SATCOM scenario. The TCP split specifically tune the TCP parameters to the context (latency, available capacity) that is measured.

One important advantage of a TCP split solution is that it does not require any end-to-end modifications and is independent for both client and server sides. That being said, this comes with a drawback: TCP splitters can hardly embed the most recent end-to-end improvements (e.g. ECN or TCP Fast Open support).

5. End-to-end solution

This section proposes an improvement of the initialization of 0-RTT connections over satellite communication where the client recalls 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] which has been shown to improve performance both for high BDP and more common BDP [CONEXT15][ICC16].

5.1. Description of the extension in the NewSessionTicket

A new extension named "BDP_data" is defined for NewSessionTicket. It contains the following value: BDP_value, that is the value in bits (same unit as [RFC6349]). The reception of the field BDP_data provides the client with 3 indications:

- The path with this server has a large BDP;
- The server added the path characteristics in the opaque 'ticket' field;
- The server will optimize the reopening of the session upon reception of this opaque ticket.

5.2. Usage of the extension in the NewSessionTicket

A server measures a connection BDP far larger than usual. It includes the path characteristics in the opaque ticket it sends to the client in a NewSessionTicket message. The message includes an additional 'extensions' field named 'BDP_data'. The client stores the session ticket and the 'BDP_data' field.

When the client reconnects to this 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_data' field (The client does not send the 'BDP_data' field back to the server). The server receives the session ticket and extracts the BDP context. It uses this information to provide a throughput closer to the capacity of the path.

As the validity of the path characteristics may change over the time the server sets the age of the ticket (see section 4.2.11.1 of [RFC8446]) to a short duration or updates the ticket when the path characteristics of the current connection changes.

6. 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 it is opaque to the client. The presence of the "BDP_data" extension field in the NewSessionTicket informs the client that the server will actively take action to improve its throughput when the session will restart.

Here are examples of 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.

- client aware of the high BDP: The 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". On top of other transport parameters used by the server, knowing that the BDP is high let the client adapt parameters specifically. As example, the client could adapt the ACK ratio following the discussion in Issue 1978 of the GITHUB repository.

- PMTU: The knowledge of the MTU of the 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 the previous connection RTT improves the throughput because the server can safely improve the slow start: e.g. using 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 later; as an example it reduces the validity to one day.

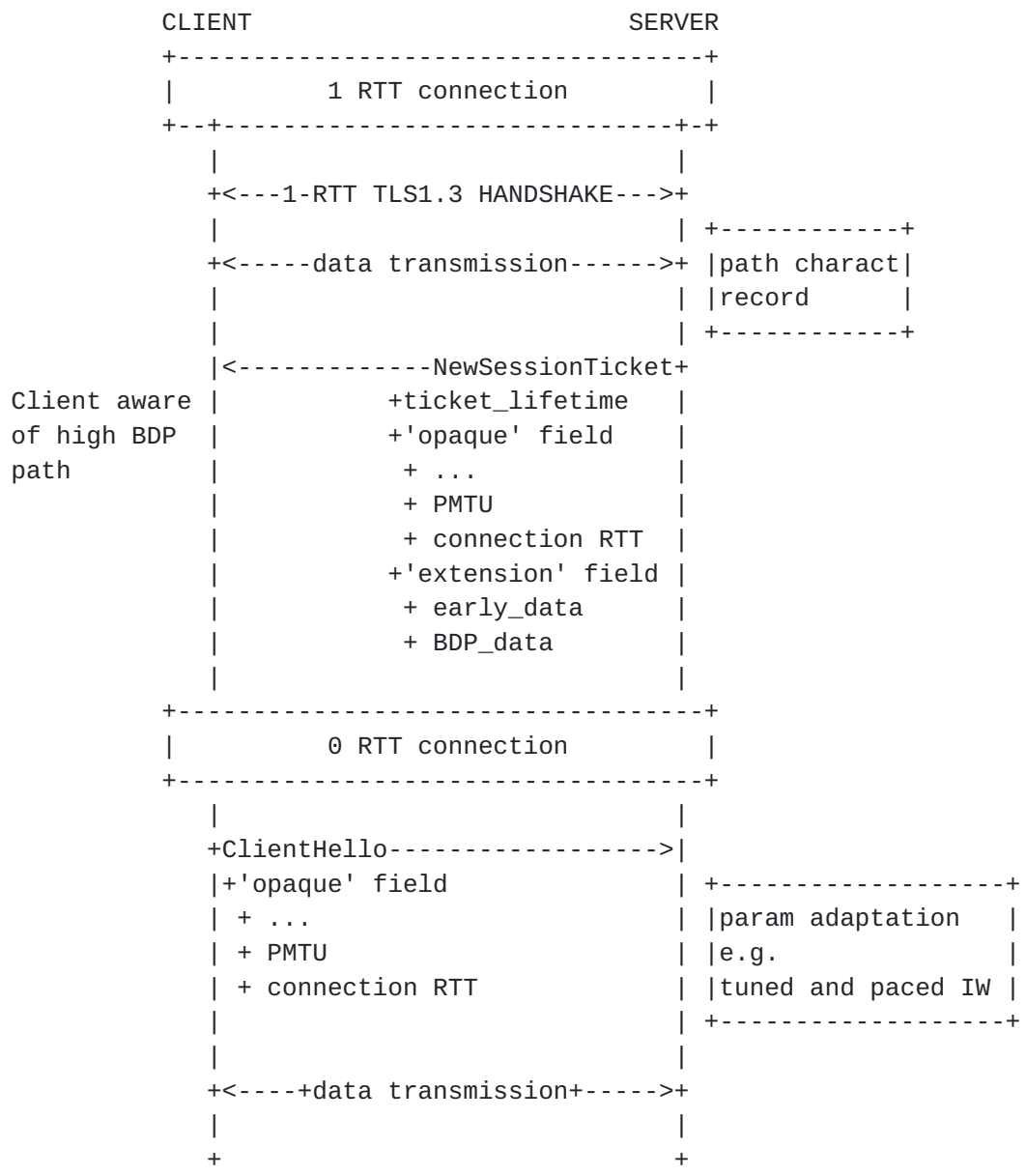

Figure 1: Example of opaque ticket content

7. Discussion

The proposal made in this draft 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_data' proposal aims at improving its ingress traffic. Improving the ingress traffic of an end user can result in drastic quality-of-experience improvements. As example, this contribution enables the exploitation of the RTT, PMTU and BDP to adapt the initial data transmission of a 0-RTT connection to halve the page load time of a web page download.

8. Acknowledgements

None.

9. Contributors

None.

10. IANA Considerations

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

11. 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_data' field is visible by the client only. An client which does not trust the server transport adaptation ignores any session ticket associated to a 'BDP_data' field.

12. References

12.1. 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>.

12.2. 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-18 (work in progress), January 2019.

[I-D.ietf-quic-tls] Thomson, M. and S. Turner, "Using TLS to Secure QUIC", draft-ietf-quic-tls-18 (work in progress), January 2019.

[I-D.ietf-quic-transport]
Iyengar, J. and M. Thomson, "QUIC: A UDP-Based Multiplexed and Secure Transport", draft-ietf-quic-transport-18 (work in progress), January 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.

[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
18 Avenue Edouard Belin
Toulouse 31400
France

Email: nicolas.kuhn@cnes.fr

Emile Stephan (editor)
Orange
2, avenue Pierre Marzin
Lannion 22300
France

Email: emile.stephan@orange.com