

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: 2 September 2026

M. Blanchet
Viagenie
W. M. Eddy
Aalyria Technologies
1 March 2026

QUIC Profile for Deep Space
draft-many-tiptop-quic-profile-02

Abstract

Deep space communications involve long delays (e.g., Earth to Mars is 4-20 minutes) and intermittent communications, because of orbital dynamics. In this context, typical QUIC stacks default transport parameters for terrestrial Internet are not suitable for deep space. This document defines a QUIC profile for deep space. It provides guidance on how to estimate and set transport parameters, advice to space mission operators and application developers on how to configure QUIC for the deep space use case and guidance to QUIC stack developers to properly expose the required transport parameters in their API.

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 2 September 2026.

Copyright Notice

Copyright (c) 2026 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document.

Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	2
1.1. Example Scenario	3
1.2. Requirements Language	3
2. RTT Calculation	3
3. Bandwidth-Delay Product(BDP) Calculation	4
4. Transport Parameters	4
4.1. Initial RTT	4
4.2. Idle Timeout	5
5. Congestion Control	5
5.1. Window Size	7
6. Flow Control	7
6.1. Max Data	7
7. Path MTU discovery	7
8. Acknowledgement Frequency	8
9. Packet Size and Sending Pace	8
10. New connection IDs	9
11. FEC	9
12. Moon Deployment Considerations	9
13. Intermittence Awareness	10
14. Application Considerations	10
15. IANA Considerations	10
16. Security Considerations	10
17. References	10
17.1. Normative References	10
17.2. Informative References	11
Acknowledgements	13
Authors' Addresses	13

1. Introduction

Deep space communications involve long delays, such as Earth to Mars is 4-20 minutes, and intermittent communications, because of orbital dynamics, such as when an orbiter is passing over a rover every 6 hours for a duration of 15 minutes.

Typical QUIC stacks default transport parameters for terrestrial Internet assume low latency such as 100-200 ms, and relative continuous connectivity. Therefore, parameters such as `initial_rtt`, `maximum_idle_timeout` have defaults typically not suitable for deep space.

Space missions are scheduled in advance and parameters such as the maximum round-trip time or bandwidth are known and determined in advance. Given relative low bandwidth in space and the intermittent communications, bandwidth usage is very precious and therefore any unneeded communication should be minimized as much as possible. Further discussion on use cases and requirements for deep space IP is discussed in [I-D.ietf-tiptop-usecase].

As described in [I-D.many-tiptop-ip-architecture], packets will be stored at either layer 2 or layer 3 by orbiters during the periods where connectivity to the next hop is not possible.

1.1. Example Scenario

To better illustrate the implication on various transport parameters, an example scenario is provided in this section.

A rover on the Mars surface is connected to a Mars surface IP network which receives intermittent connectivity from a few orbiters with an average of 6 hours per orbit. Some of those orbiters have circular orbits, other elliptical. The latter means that the overpass are not at a fixed frequency. The orbiters are connected to Earth ground station while they are in line of sight with Earth. Earth and Mars have variable distance from 4 to 20 minutes light seconds. That one way delay however change "slowly" as the planets are orbiting around the Sun.

1.2. Requirements Language

The key words "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.

2. RTT Calculation

A QUIC stack estimates the round-trip time(RTT) between the two peers over the period of the connection. This is used for example to initiate the retransmission of packets when the acknowledgement of those packets is not received within the expected RTT. Using the example in Section 1.1, it is necessary to prime the QUIC stack with the right initial values, to avoid, for example, to retransmit packets after 100 ms while the expected RTT is 2 hours.

A space application designer should calculate the maximum RTT for its mission. Using the example in Section 1.1, the maximum RTT due to the maximum two-way delay is 45 minutes and the one due to the overpass frequency is 6 hours, therefore the maximum RTT is 6 hours 45 minutes.

A space application designer should calculate the minimum RTT for its mission. Using the example in Section 1.1, the minimum RTT due to the minimum two-way delay is 8 minutes and the one due to the overpass frequency is 0 assuming direct line of sight for the whole path, therefore the minimum RTT is 8 minutes.

3. Bandwidth-Delay Product(BDP) Calculation

A QUIC stack like any transport stack manages the pacing of sending packets from the source to avoid overloading the network, creating congestion and to avoid overloading the other peer.

A space application designer should calculate the bandwidth-delay product(BDP) of the whole path for its mission. The minimum BDP should be calculated with the minimum RTT and the minimum bandwidth used during those times. The maximum BDP should be calculated with the maximum RTT and the maximum bandwidth.

4. Transport Parameters

This section discusses how to configure some QUIC transport parameters. The parameter names are from the IANA registry[iana_quic_transport].

4.1. Initial RTT

To prime the QUIC stack with the expected RTT of the mission, an application should set the Initial RTT on connection establishment to the maximum RTT as calculated in Section 2.

If the set RTT is too low, then retransmission will be sent before the actual acknowledgement was received. In this case, the QUIC stack will still converge and deliver reliable data, but at the expense of extra bandwidth used. If the set RTT is too high, then when a packet is lost, the retransmission will be started later than the optimal time, therefore the total time to transmit all the data, including the losses recovered, will be longer than if it was set properly, but the QUIC stack will still converge and deliver reliable data.

The `initial_rtt` transport parameter is specified in [RFC9002].

An application may use the resume mechanism as described in [I-D.ietf-tsvwg-careful-resume] to update the RTT during the connection lifetime.

4.2. Idle Timeout

To avoid the QUIC stack to terminate a connection due to no activity from the other peer, an application should set the Idle Timeout on connection establishment to the maximum RTT as calculated in Section 2.

If the set RTT is too low, then the other peer may terminate the connection before all the data is received. In this case, the QUIC stack on the sender side will need to reestablish the connection, possibly using the ORTT mechanism, and resend the data that was not acknowledged previously. In this case, the application shall still recover and provide full data reliability but at the expense of more total time and extra bandwidth used. If the set RTT is too high, then the other peer will close its side of the connection later than needed in the event of a lost connection. In this case, the resources used by keeping the connection, such as memory, will not be released as fast as it could be if the RTT was properly set.

The `max_idle_timeout` transport parameter is specified in section 8.2 of [RFC9000].

5. Congestion Control

In the Internet stack, congestion control (CC) in transport protocols is crucial to sharing network resources among concurrent traffic flows. Minimizing congestion is necessary in order to prevent unnecessary queueing latency and packet drops. Problems with typical Internet congestion control algorithms in deep space settings can include: (1) assumptions of timely in-band feedback/information, (2) interpretation of large delays as congestion/loss, and (3) lack of knowledge about overall network state and management decisions.

Internet transport protocols use estimates of loss events, round-trip latency, and Explicit Congestion Notification (ECN) [RFC3168][RFC8087][RFC8311] to sense connection. All of these methods operate in a "closed loop" fashion on the scale of RTTs, in order to react responsively in normal Internet conditions. When the RTTs are many minutes or more in some deep space scenarios, this is not effective in quickly detecting and reacting to congestion. Instead of relying on in-band closed loop detection and response to congestion, deep space congestion control could be assisted by management/control plane information, as deep space access, networking, and mission operations are substantially coordinated, in contrast to normal Internet access.

[I-D.many-tiptop-ip-architecture] discusses that given intermittent connectivity in deep space, nodes may need to temporarily store either L2 frames or L3 packets when links are down until the link is up again. This behavior will be interpreted by various CC algorithms as congestion. However, this is not necessarily the right behavior, since the delays generated are due to waiting for a lower-layer transmission opportunity, but not necessarily lack of capacity when it becomes available to transmit.

Deep space networking differs from traditional Internet use, because it is highly coordinated. Antennas must be pointed and track physically, and operations centers must work together between agencies/companies/etc in order to manage resources (including spectrum usage, physical systems, compute and storage, etc.) and orchestrate configurations and activities over time. Planning and scheduling is a significant activity, and results in the ability to understand traffic flow needs and allocate capacity for application flows, unlike in the traditional Internet.

Other space-oriented transport protocols, such as SCPS-TP, have included rate-based open loop congestion control algorithms. These algorithms send at pre-determined rates, without necessarily requiring in-band feedback, with the rate information over time provided by variables managed through an external scheduling system or other means.

Therefore, QUIC stacks for deep space should be configurable to use rate-based open-loop congestion control, that allows relevant configuration variables such as window, pacing rate, expected RTT, and others to be governed by schedule-driven management/control plane cues, rather than closed loop in-band probing/estimation.

5.1. Window Size

A QUIC stack manages the pacing of the source by the window size. A typical value used for Internet is 2 times the BDP. In space, careful considerations must be taken. A too low BDP means that the source node may not be sending enough packets to completely use the network and the available bandwidth of the links, which is less optimal given the scarcity of communications in space. Therefore, an application should not use a BDP lower than the minimum BDP as calculated in Section 3. A too large BDP may use too much of the bandwidth of the links.

Since packets may be stored at either layer 2 or layer 3 by intermediate nodes, the maximum storage of in-flight packets in these intermediary nodes is to be considered. Therefore, space operations should properly identify the best window size based on the minimum and maximum BDP and storage size of the intermediary nodes for the mission/application. As those parameters are known in advance for a mission, these can be set appropriately on connection establishment by the application.

6. Flow Control

6.1. Max Data

`initial_max_data` is the maximum number of bytes that can be sent on a connection [RFC9000]. `initial_max_stream_data` is similar but per stream. Given the BDP of a typical deep space connection, applications should set these parameters to enough large values so that the source is capable of sending data while the bandwidth is available.

The various `initial_max_data` transport parameters are specified in section 8.2 of [RFC9000].

7. Path MTU discovery

To find the optimum MTU, some QUIC stacks implement Path MTU discovery [RFC8899], which sends bigger packets every time until it discovers the maximum MTU, which may involve packet loss. Given that in deep space, MTUs for all links on the paths may be known in advance and that probing is inefficient and not timely, the application developer may elect to disable the path MTU discovery mechanism and set the real path MTU on connection establishment of the application.

For instance, a 1280 byte path MTU might be assumed, based on knowledge of the network design and assurance from network providers that this would be supported without fragmentation, after accounting for any encapsulation present on the scheduled paths.

Given coordination between the small number of network operators involved in an end-to-end path, support for large MTUs might be coordinated and known in advance. This could be desirable for efficiency, but is out of scope of this document.

Since scheduled paths may change over time, to enable use a known optimal value at any time, the path MTU can also be exposed by an implementation as a dynamically managed parameter, set according to a schedule.

8. Acknowledgement Frequency

QUIC stacks have various mechanisms to trigger acknowledgements (ACKs), as described in [RFC9000], [I-D.ietf-quic-ack-frequency]. There are advantages of sending "frequent" acknowledgements, such as the source can free out memory of received frames earlier. However, this uses more capacity for sending ACKs, and could be problematic in cases of very high link bandwidth or network path capacity asymmetry.

Adjusted ACK frequency information is provided to a QUIC peer through an in-band signal in a QUIC ACK_FREQUENCY frame extension to the QUIC base specification. Usage needs to be negotiated via transport parameters on connection establishment, and desired values need to be provided (e.g. directly by an application or supplied to the stack via a management system). The `max_ack_delay` and `ack_delay_exponent` transport parameters are specified in section 8.2 of [RFC9000], and requested `max_ack_delay` changes are provided in ACK_FREQUENCY frames.

The optimal ACK frequency is related to the bandwidth asymmetry between the directions of data and acknowledgement flow. This may vary over time, but also can be known in advance by a route orchestrator or other management system. In order to optimize performance, ACK frequency may be exposed as a dynamically controlled or time-varying parameter.

9. Packet Size and Sending Pace

There are pros and cons of sending large packets. Sending smaller packets means using more bandwidth because of multiple headers, unless header compression is used, but decreases the probability of packet loss which in space should be minimized. Sending larger packets means more efficient use of bandwidth, but in front of significant packet loss, may in fact result in using more bandwidth

than smaller packets, since they will have to be re-transmitted.

In the context of Internet, QUIC stacks may elect to not wait much time in order to add more frames in a single packets to deliver faster to the receiving end point.

For deep space applications, where time for transmission is many orders of magnitude longer than on Internet, a QUIC stack may be configured to wait "a bit more" to add more frames to a single packet. For example, before sending a packet, a QUIC stack may wait to process all incoming packets in case the latter may elect to add frames on the response packet.

10. New connection IDs

QUIC stacks typically preemptively send new connection IDs to the other peer, for future needs such as future connection migration. However, those use cases may not be happening often in deep space. That optional optimization of sending new connection IDs may not be needed for deep space use case, while the actual cost of these additional bytes is pretty low.

11. FEC

CCSDS deep space links uses FEC at layer 2 (TODO: add ref to CCSDS book), using a pipeline of codecs, enabling low frame error rate in the presence of a higher signal bit error rate. While FEC for QUIC has been defined [I-D.michel-quic-fec], it remains to be seen if it is really needed for deep space.

12. Moon Deployment Considerations

Earth to Moon is just a few light seconds away. When the whole path is all line of sight, it is possible to use QUIC stacks as configured today, but it will take more time to converge, therefore less optimal. The BBR algorithm will be a better choice as it is based on delay to measure congestion.

However, if one wants to consider orbiters that will have intermittent communications, then the scenario discussed in Section 1.1 also applies and calculating RTT and BDP as discussed previously apply.

13. Intermittence Awareness

Another way to solve the generic problem is to make transport aware of the intermittence periods, so that when there is a direct path end-to-end without any intermittence, the normal QUIC behavior such as congestion control may be used with proper RTT configuration, and then a different behavior in the context of intermittence. However, the actual scheduling of communication windows is pretty complicated and have a lot of variations that an intermittence-aware transport will be very fragile.

14. Application Considerations

On terrestrial Internet, the cost, by time or bandwidth, to re-establish a connection to the same peer is very low, since it is re-established very fast and without too much use of bandwidth. Therefore, often, applications are designed in a pattern of establishing a connection, sending a query, getting the response, closing the connection and redoing the whole process for the next query. In deep space, the cost, specially the delay, of re-establishing a connection is very high. Therefore, applications should be carefully designed to not close connections after a query-response, if there is a possible new queries in the future.

15. IANA Considerations

This memo includes no request to IANA.

16. Security Considerations

The use of 0-RTT is subject to replay attacks[RFC9001] and therefore should be considered to be disabled depending on the security policy of the mission.

Certificates and keys need to be renewed before their expiration, taking into account the delay to send, receive and confirm. Protocols such as OCSP[RFC6960] providing on-line real-time validation and revocation check will likely not work given the too long delays, therefore certificates need to be validated using local trust anchors.

The use of long term keys, such as ones set prior to launch, may create exposure, therefore keys should be renewed at appropriate frequency.

17. References

17.1. 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>>.
- [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>>.

17.2. Informative References

- [RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168, September 2001, <<https://www.rfc-editor.org/info/rfc3168>>.
- [RFC6960] Santesson, S., Myers, M., Ankney, R., Malpani, A., Galperin, S., and C. Adams, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP", RFC 6960, DOI 10.17487/RFC6960, June 2013, <<https://www.rfc-editor.org/info/rfc6960>>.
- [RFC8087] Fairhurst, G. and M. Welzl, "The Benefits of Using Explicit Congestion Notification (ECN)", RFC 8087, DOI 10.17487/RFC8087, March 2017, <<https://www.rfc-editor.org/info/rfc8087>>.
- [RFC8311] Black, D., "Relaxing Restrictions on Explicit Congestion Notification (ECN) Experimentation", RFC 8311, DOI 10.17487/RFC8311, January 2018, <<https://www.rfc-editor.org/info/rfc8311>>.
- [RFC8899] Fairhurst, G., Jones, T., T端 xen, M., R端 ngeler, I., and T. V端 lker, "Packetization Layer Path MTU Discovery for Datagram Transports", RFC 8899, DOI 10.17487/RFC8899, September 2020, <<https://www.rfc-editor.org/info/rfc8899>>.
- [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>>.
- [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>>.
- [I-D.michel-quic-fec]
Michel, F. and O. Bonaventure, "Forward Erasure Correction for QUIC loss recovery", Work in Progress, Internet-Draft, draft-michel-quic-fec-01, 23 October 2023, <<https://datatracker.ietf.org/doc/html/draft-michel-quic-fec-01>>.
- [I-D.ietf-tsvwg-careful-resume]
Kuhn, N., Stephan, E., Fairhurst, G., Secchi, R., and C. Huitema, "Convergence of Congestion Control from Retained State", Work in Progress, Internet-Draft, draft-ietf-tsvwg-careful-resume-24, 1 October 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-tsvwg-careful-resume-24>>.
- [I-D.ietf-quic-ack-frequency]
Iyengar, J., Swett, I., and M. K^端hlewind, "QUIC Acknowledgment Frequency", Work in Progress, Internet-Draft, draft-ietf-quic-ack-frequency-14, 5 February 2026, <<https://datatracker.ietf.org/doc/html/draft-ietf-quic-ack-frequency-14>>.
- [I-D.ietf-tiptop-usecase]
Blanchet, M., Eddy, W., and M. Eubanks, "IP in Deep Space: Key Characteristics, Use Cases and Requirements", Work in Progress, Internet-Draft, draft-ietf-tiptop-usecase-01, 21 January 2026, <<https://datatracker.ietf.org/doc/html/draft-ietf-tiptop-usecase-01>>.
- [I-D.many-tiptop-ip-architecture]
Blanchet, M., Eddy, W., and T. Li, "An Architecture for IP in Deep Space", Work in Progress, Internet-Draft, draft-many-tiptop-ip-architecture-02, 29 September 2025, <<https://datatracker.ietf.org/doc/html/draft-many-tiptop-ip-architecture-02>>.
- [iana_quic_transport]
Internet Assigned Numbers Authority(IANA), "QUIC Transport Parameters", September 2025, <<https://www.iana.org/assignments/quic/quic.xhtml#quic-transport>>.

Acknowledgements

This document and its underlying work has been reviewed and discussed by many, who have provided valuable feedback and comments, including disagreements, and made an overall more solid document. These people are, in no specific order: Lars Eggert, Christian Huitema, Adolfo Ochagavia, Mirja Kuehlewind, Michael Richardson.

The Quinn QUIC stack was used for testing. We would like to acknowledge the help of Benjamin Saunders and Adolfo Ochagavia in using Quinn.

Authors' Addresses

Marc Blanchet
Viagenie
Canada
Email: marc.blanchet@viagenie.ca

Wesley M. Eddy
Aalyria Technologies
United States of America
Email: wes@aalyria.com