TCP Maintenance and Minor Extensions
Internet-Draft
Intended status: Standards Track
Expires: 8 January 2026

M. Baerts
UCLouvain & NGI Zero Core

7 July 2025

Multipath TCP with longer DSS mappings
draft-baerts-tcpm-mptcpdss-00

Abstract

This document proposes an extension to improve Multipath TCP based on operational experience by allowing Multipath TCP to use DSS mappings that are longer than 64 KBytes.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at https://ipnetworkinglab.github.io/draft-mptcp-dss/draft-baerts-tcpm-mptcpdss.html. Status information for this document may be found at https://datatracker.ietf.org/doc/draft-baerts-tcpm-mptcpdss/.

Discussion of this document takes place on the TCP Maintenance and Minor Extensions Working Group mailing list (mailto:tcpm@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/tcpm/. Subscribe at https://www.ietf.org/mailman/listinfo/tcpm/.

Source for this draft and an issue tracker can be found at https://github.com/IPNetworkingLab/draft-mptcp-dss.

Status of This Memo

Table of Contents

1. Introduction

From a performance viewpoint, TCP stacks are optimised to leverage large segments and use TCP Segment Offload / Generic Receive Offload (TSO/GRO). The DSS option defined in Multipath TCP allows to map a series of bytes from the bytestream on a specific subflow. Unfortunately, the length of this mapping is encoded in a 16-bit field. Since each Multipath TCP segment must include a DSS mapping before being sent to the network interface, this restricts the size of the segments that Multipath TCP can use. In particular in IPv6, it is impossible for Multipath TCP to leverage IPv6 jumbograms [RFC2675] in contrast to regular TCP. This document proposes a modification of the DSS option to support longer mappings.

2. Conventions and Definitions

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.

3. Extending the DSS option

The extension proposed in this document is pretty simple. Given that the DSS Checksum is rarely used in practice, we propose to reuse the space reserved for this checksum in the DSS option to support 32-bit data-level mappings. This enables Multipath TCP servers to send segments that are longer than 64 KBytes. This extension is negotiated using the TBD flag in the MP_CAPABLE option during the handshake.

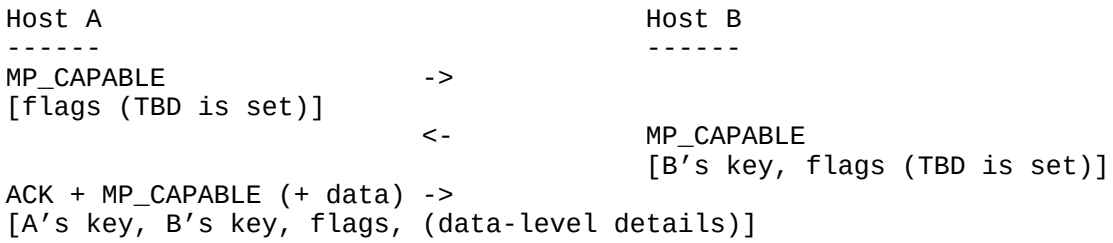

Figure 1: Negotiation of the DSS option with longer mappings

The DSS option defined in [RFC8684] reserves 16 bits for the Checksum. However, operational experience indicates that this checksum is almost never used by Multipath TCP deployments. It was designed to detect middlebox interference caused notably by Application Level Gateways that modify TCP payloads [RFC8041]. Given the widespread adoption of TLS, such ALGs are rarely used by applications using Multipath TCP [MPTCP-longitudinal].

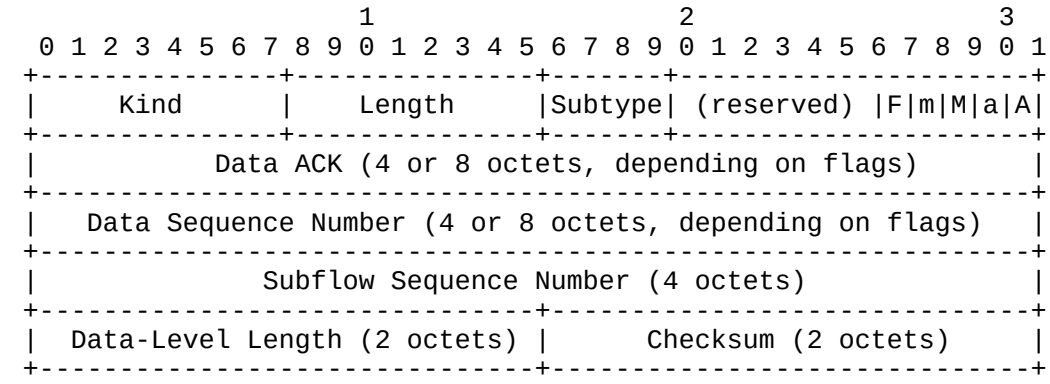

Figure 2: The DSS option in RFC8684

This document proposes to use a 32-bit Data-Level Length to support large TCP segments. The new DSS option is shown in Figure 3. The other fields of this option and the procedures defined in [RFC8684] are unchanged.

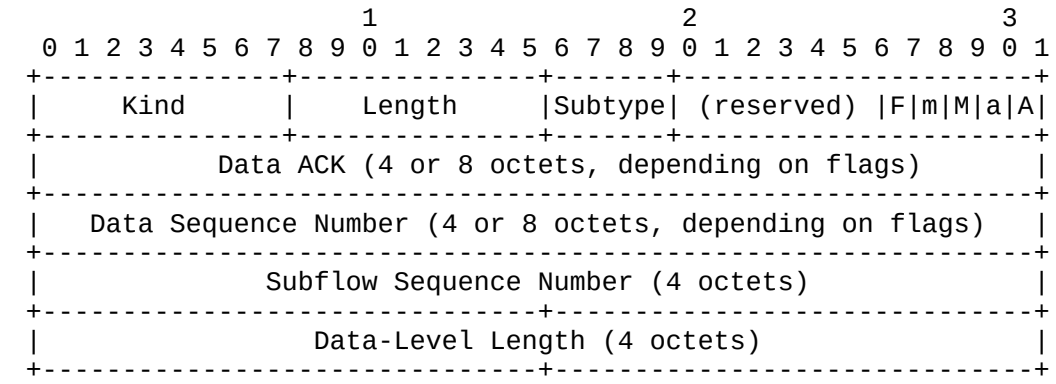

Figure 3: The new DSS option

[RFC8684] defines the MP_CAPABLE option as shown in Figure 4. This option contains several flags, A-H. Flags A, B, C, and H are specified in [RFC8684]. This document uses Flag TBD to indicate in a SYN that the initiator of a connection requests the utilization of 32-bit Data-Level Length. If this Flag is set in a SYN, Flag A must also obviously be set to 0 to indicate that the Checksum is not required on this connection. If both Flags A and TBD are set in a SYN, the receiver MUST not continue the MPTCP connection, and SHOULD fallback to TCP. A server that receives a SYN with the TBD Flag set can reply with:

* a SYN+ACK with the TBD Flag set to 1 to confirm that it accepts to use 32-bit Data-Level Length

* a SYN+ACK with the TBD Flag set to 0 to indicate that it prefers to use 16-bit Data-Level Length

Even when the TBD Flag is set to 1, the MP_CAPABLE options continue to use a 16-bit Data-Level Length like before, to allow fallback if the receiver doesn’t support a 32-bit Data-Level Length.

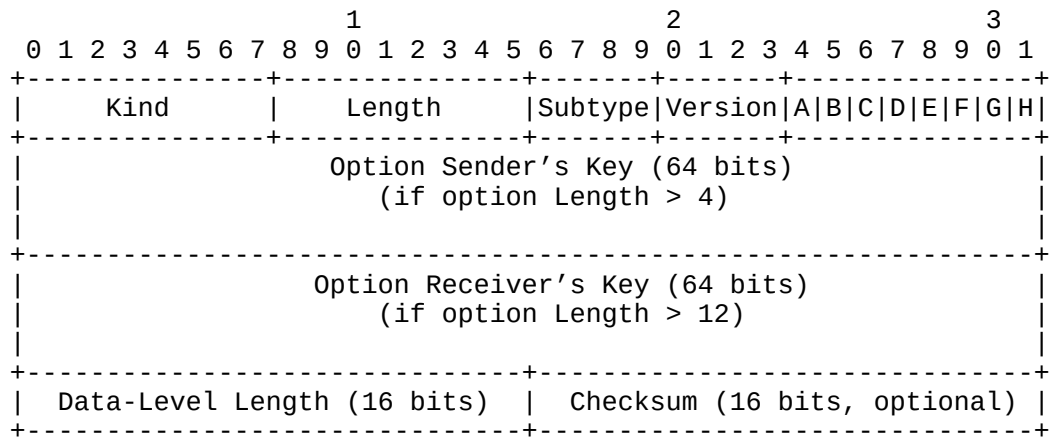

Figure 4: The MP_CAPABLE option

4. Security Considerations

This document does not change the security considerations defined in [RFC8684].

5. IANA Considerations

This document requests the IANA to reserve flag TBD of the MP_CAPABLE option as defined in this document. It also proposes to change the format of the DSS option. This document suggests using the D flag of the MP_CAPABLE option.

6. References

6.1. 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/rfc/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/rfc/rfc8174>.

[RFC8684] Ford, A., Raiciu, C., Handley, M., Bonaventure, O., and C. Paasch, "TCP Extensions for Multipath Operation with Multiple Addresses", RFC 8684, DOI 10.17487/RFC8684, March 2020, <https://www.rfc-editor.org/rfc/rfc8684>.

6.2. Informative References

[MPTCP-longitudinal]
Shreedhar, T., Zeynali, D., Gasser, O., Mohan, N., and J. Ott, "A Longitudinal View at the Adoption of Multipath TCP", arXiv, DOI 10.48550/ARXIV.2205.12138, 2022, <https://doi.org/10.48550/ARXIV.2205.12138>.

[RFC2675] Borman, D., Deering, S., and R. Hinden, "IPv6 Jumbograms", RFC 2675, DOI 10.17487/RFC2675, August 1999, <https://www.rfc-editor.org/rfc/rfc2675>.

[RFC8041] Bonaventure, O., Paasch, C., and G. Detal, "Use Cases and Operational Experience with Multipath TCP", RFC 8041, DOI 10.17487/RFC8041, January 2017, <https://www.rfc-editor.org/rfc/rfc8041>.

Acknowledgments

This project is funded through NGI Zero Core, a fund established by NLnet with financial support from the European Commission’s Next Generation Internet program.

Author’s Address

Matthieu Baerts
UCLouvain & NGI Zero Core
Email: matttbe@kernel.org