Payload Protocol Identifier based Fragmentation and Reassembly for the Stream Control Transmission Protocol

Abstract

This document describes a method for the Stream Control Transmission Protocol (SCTP) allowing the upper layer to perform fragmentation, reassembly, and interleaving of large ordered user messages by using the payload protocol identifier (PPID).

According to the base specification supporting fragmentation of large user messages is optional. And even if an SCTP implementation supports fragmentation, interleaving of user messages is not supported by the base specification.
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. Conventions
3. Sender Side Considerations
4. Receiver Side Considerations
5. Socket API Considerations
6. IANA Considerations
7. Security Considerations
8. References
   8.1. Normative References
   8.2. Informative References
Authors' Addresses

1. Introduction

This document specifies a method to use PPIDs for fragmenting ordered large user messages. Using this method also allows the ability to interleave large user messages as provided by [RFC8260] in combination with using the SCTP_FRAGMENT_INTERLEAVE level_2 as described [RFC6458], Section 8.1.20.

Reasons to use this method include:

*The fragmentation of large user messages is only an optional feature of SCTP implementations compliant [RFC9260]. Therefore, if an implementation does not support fragmentation, it is impossible to send large user messages requiring fragmentation.

*An SCTP implementation supporting [RFC9260], but not [RFC8260], does not allow the interleaving of large user messages.

This method does not apply to user messages sent using partial reliability as described in [RFC3758].

The idea described in this document was already described in [I-D.ietf-rtcweb-data-channel]. In the final specification [RFC8831], this method is declared deprecated.

2. Conventions

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. Sender Side Considerations

An upper layer splits a user message in one or more user message fragments. The upper layer SHOULD choose the size of the user message fragments such that SCTP level fragmentation is avoided.

The upper layer uses two PPIDs. It MUST use one PPID for all user messages fragments except the last one, and it MUST the other PPID for the last user message fragments.

All user message fragments belonging to the same user message MUST be sent on the same stream, reliable, and ordered in the sequence they belong to the user message. User message fragments sent on different stream MAY be sent in any order. This allows the interleaving of user messages sent on different streams.

User messages not requiring to be split into multiple user message fragments are sent as a single user message fragment with the PPID used for last user fragments.

4. Receiver Side Considerations

The upper layer MUST process user message fragments received on different streams independently. All user message fragments are received by the upper layer in the correct ordering and the PPID MUST be used to reconstruct the user message boundaries. A user message fragment with the PPID marking the last user message fragment is the last fragment of a use message. The next received user message fragment on the stream is the first fragment of the next user message.

An upper layer MUST deal with interleaving of user messages.

Please note that notifications, if enabled, can be provided by the SCTP implementation at any time.

5. Socket API Considerations

This document does not require and changes or additions to the Socket API described in [RFC6458].

6. IANA Considerations

This document does not make any requests for IANA.
7. Security Considerations

This document does not change the considerations given in [RFC9260].

8. References

8.1. Normative References


8.2. Informative References


Authors' Addresses

Michael Tüxen
Münster University of Applied Sciences
Stegerwaldstrasse 39
48565 Steinfurt
Germany

Email: tuexen@fh-muenster.de

Randell Jesup
Mozilla Corporation
1835 Horse Shoe Trl
Malvern, PA 19355
United States of America

Email: randell-ietf@jesup.org

Hannes Tschofenig

Email: hannes.tschofenig@gmx.net