

Delay/Disruption Tolerant Networking
Internet-Draft
Intended status: Standards Track
Expires: 21 December 2025

R. Taylor
Aalyria Technologies
19 June 2025

Forward Error Correction for the Bundle Transfer Protocol
draft-ietf-dtn-btpu-fec-00

Abstract

This document defines an optional extension to the Bundle Transfer Protocol - Unidirectional, as described in [BTPU], to enable forward error correction (FEC) coding to be applied selectively to the transfer of individual bundles on a case by case basis.

The definition and use of FEC follows the FECFRAME framework defined in [RFC6363], and this document introduces new Message types to BTPU in order to carry the FEC information as defined in the framework.

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://ricktaylor.github.io/btpu-fec/draft-ietf-dtn-btpu-fec.html>. Status information for this document may be found at <https://datatracker.ietf.org/doc/draft-ietf-dtn-btpu-fec/>.

Discussion of this document takes place on the Delay/Disruption Tolerant Networking Working Group mailing list (<mailto:dtn@ietf.org>), which is archived at <https://mailarchive.ietf.org/arch/browse/dtn/>. Subscribe at <https://www.ietf.org/mailman/listinfo/dtn/>.

Source for this draft and an issue tracker can be found at <https://github.com/ricktaylor/btpu-fec>.

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 21 December 2025.

Copyright Notice

Copyright (c) 2025 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	3
1.1. Applicability	3
2. Conventions and Definitions	3
3. Protocol Overview	4
3.1. Pre-agreed FEC Instance ID	4
4. Message Definitions	5
4.1. Pre-agreed FEC Source Message	5
4.2. Explicit FEC Source Message	6
4.3. Pre-agreed FEC Repair Message	7
4.4. Explicit FEC Repair Message	7
5. Security Considerations	8
6. IANA Considerations	8
7. References	8
7.1. Normative References	8
7.2. Informative References	9
Appendix A. Acknowledgments	9
Author's Address	9

1. Introduction

There are a number of use-cases of the Bundle Transfer Protocol - Unidirectional [BTPU], where the use of transfer segment repetition as a mechanism to protect against the loss of frames can be considered sub-optimal. This document describes an alternate mechanism based on forward error correction (FEC) coding, that requires increased computational complexity but fewer transmitted bits.

Rather than defining novel formats and registries for the variety of standardized FEC mechanisms, this document reuses the primitives and best practices of the FECFRAME framework, defined in [RFC6363].

Just as in core BTPU, a Bundle is split in a series of octet sequences that are emitted into Messages by the sender to be transported to receivers by the underlying link-layer protocol; but when FEC is desired, the mechanisms defined in the FECFRAME framework are used to produce a sequence of Source Blocks and Repair Symbols that are placed into new Messages, rather than just sub-slices of the original Bundle. The new Messages are used to distinguish FEC Source Blocks and FEC Repair Symbols from core BTPU Segments.

Although the content and processing of the new Messages differs from existing BTPU Messages, the rules around the emission and replication of the Messages are identical to the rules applicable to the core BTPU Segment Messages, and they follow the common BTPU Message format, allowing implementations that do not support this extension to efficiently detect and ignore the new Messages.

1.1. Applicability

It should be noted that when FEC is available at the link-layer it is generally more effective than applying it at the Transfer layer, and should probably be used when it is available. This extension is designed to provide FEC capabilities when the underlying link-layer protocol does not have native support for FEC, or when per-Transfer FEC is desired by a deployment.

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. Protocol Overview

Rather than updating the Segment Messages defined in BTPU, this extension introduces two new pairs of Messages to carry the source and repair symbols of a BTPU Bundle protected with FEC. The use of new types allows a deployment to select the use of FEC as appropriate on a per-transfer basis, perhaps associated with some upper layer concept of reliability for a particular transfer, or change in transmission environment.

In the language of Section 2 of [RFC6363], the FEC Source Messages act as FEC Source Packets, and the FEC Repair Messages as FEC Repair Packets. Within the context of a particular Transfer, the sequence of FEC Source Messages are considered the Source Flow, and the sequence of FEC Repair Messages the Repair Flow.

The source and repair Messages are grouped into two pairs:

Pre-agreed FEC: The Pre-agreed FEC Source (Section 4.1) and Pre-agreed FEC Repair (Section 4.3) Messages provide a wire-efficient format, for use when the FEC Framework Configuration Information Section 5.5 of [RFC6363] has been pre-agreed via some a-priori configuration or out-of-band mechanism.

Explicit FEC: The Explicit FEC Source (Section 4.2) and Explicit FEC Repair (Section 4.4) Messages include the FEC-Scheme-Specific Information (FSSI) in the Message content, allowing for the ad-hoc use of different FEC schemes and configuration, given underlying implementation support.

Irrespective of whether Pre-agreed or Explicit FEC is in use for a Transfer, the FEC Framework Configuration Information MUST NOT change mid-transfer. If a receiver detects a change in FEC Framework Configuration Information during a Transfer, it MUST consider any incomplete Transfer affected by the change as cancelled, and ignore all future Messages associated with the Transfer.

3.1. Pre-agreed FEC Instance ID

When pre-agreed FEC is desired, a lookup table MUST be configured at the sender and all receivers that maps a unique identifier, the FEC Instance ID, to a particular FEC scheme and corresponding FSSI, such that each Pre-agreed FEC Source (Section 4.1) and Pre-agreed FEC Repair (Section 4.3) Message can refer to the FEC mechanism in use by referencing the FEC Instance ID, rather than including all the FEC configuration information in each Message.

The FEC Instance ID is an unsigned integer in the range 0..255 inclusive, and is carried in the respective FEC Messages encoded in the FEC Instance ID field. Just like the FEC scheme and configuration, the FEC Instance ID MUST be the same for all Messages concerned with an individual Transfer. If a receiver detects a change in FEC Instance ID during a Transfer, it MUST consider the Transfer cancelled, and ignore all future Messages associated with the Transfer.

Configuration of the mapping of FEC Instance ID to FEC scheme information MUST be performed out-of-band, or via a-priori configuration mechanism.

4. Message Definitions

All new Messages introduced in this document follow the common message format as defined in Section 4 of [BTPU].

This specification deviates from the recommendation in Section 5.3 of [RFC6363] by placing the Explicit Source FEC Payload ID before the Source Data, as BTPU has no capability analogous to common header compression, as found in Robust Header Compression (ROHC) [RFC3095], and therefore to maintain consistency with other BTPU messages, the metadata precedes the data.

4.1. Pre-agreed FEC Source Message

The Pre-agreed FEC Source Message is used to encapsulate a Source Block (Section 2 of [RFC6363]) of a Bundle Transfer that uses FEC with a pre-agreed configuration.

A Pre-agreed FEC Source Message has a type of TBD1. The Message Content field is formatted as follows:

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
| Transfer Number                                                                 |
+-----+-----+-----+-----+-----+-----+-----+-----+
| FEC Inst. ID | Explicit Source FEC Payload ID ...                           :
+-----+-----+-----+-----+-----+-----+-----+-----+
| ... Source Block Data ...                                                     :
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Transfer Number: The numeric identifier of the Transfer that this source block is part of, encoded as a 32-bit unsigned integer in network byte order.

FEC Instance ID: The FEC Instance ID (Section 3.1) of the pre-agreed FEC scheme and configuration in use for the Transfer, encoded as an 8-bit unsigned integer in network byte order.

Explicit Source FEC Payload ID: The Explicit Source FEC Payload ID as defined in Section 5.3.1 of [RFC6363].

Source Block Data: The octets of the source block of the Transfer, with the length calculated as the Message content length excluding the length of the Transfer Number, FEC Instance ID, and Explicit Source FEC Payload ID.

4.2. Explicit FEC Source Message

The Explicit FEC Source Message is used to encapsulate a Source Block (Section 2 of [RFC6363]) of a Bundle Transfer that uses FEC with an explicit FEC scheme and configuration.

A Explicit FEC Source Message has a type of TBD2. The Message Content field is formatted as follows:

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
| Transfer Number                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
| FEC Enc. ID   | FEC-Scheme-Specific Information elements ... :
+-----+-----+-----+-----+-----+-----+-----+-----+
| Explicit Source FEC Payload ID ...                  :
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     ... Source Block Data ...
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Transfer Number: The numeric identifier of the Transfer that this source block is part of, encoded as a 32-bit unsigned integer in network byte order.

FEC Encoding ID: A FEC Encoding ID, as defined in Section 5.6 of [RFC6363], encoded as an 8-bit unsigned integer in network byte order.

FEC-Scheme-Specific Information elements: Zero or more FEC-Scheme-Specific Information elements as defined in Section 5.5 of [RFC6363].

Explicit Source FEC Payload ID: The Explicit Source FEC Payload ID as defined in Section 5.3.1 of [RFC6363].

Source Block Data: The octets of the source block of the Transfer, with the length calculated as the Message content length excluding the length of the Transfer Number, FEC Encoding ID, FEC-Scheme-Specific Information elements, and Explicit Source FEC Payload ID.

4.3. Pre-agreed FEC Repair Message

The Pre-agreed FEC Repair Message is used to encapsulate the Repair Symbols (Section 2 of [RFC6363]) of a Bundle Transfer that uses FEC with a pre-agreed configuration.

A Pre-agreed FEC Repair Message has a type of TBD3. The Message Content field is formatted as follows:

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
| Transfer Number                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
| FEC Inst. ID |                                     Repair FEC Payload ID ... :
+-----+-----+-----+-----+-----+-----+-----+-----+
|                                     ... Repair Symbol Data ... :
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Transfer Number: The numeric identifier of the Transfer that this source block is part of, encoded as a 32-bit unsigned integer in network byte order.

FEC Instance ID: The FEC Instance ID (Section 3.1) of the pre-agreed FEC scheme and configuration in use for the Transfer, encoded as an 8-bit unsigned integer in network byte order.

Repair FEC Payload ID: The Repair FEC Payload ID as defined in Section 5.4.1 of [RFC6363].

Repair Symbol Data: The octets of the repair symbols of the Transfer, with the length calculated as the Message content length excluding the length of the Transfer Number, FEC Instance ID, and Repair FEC Payload ID.

4.4. Explicit FEC Repair Message

The Explicit FEC Repair Message is used to encapsulate the Repair Symbols (Section 2 of [RFC6363]) of a Bundle Transfer that uses FEC with an explicit FEC scheme and configuration.

A Explicit FEC Repair Message has a type of TBD4. The Message Content field is formatted as follows:

```

      0               1               2               3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+
| Transfer Number                                                     |
+-----+-----+-----+-----+-----+-----+-----+-----+
| FEC Enc. ID   | FEC-Scheme-Specific Information elements ... :
+-----+-----+-----+-----+-----+-----+-----+-----+
| Repair FEC Payload ID ... :
+-----+-----+-----+-----+-----+-----+-----+-----+
|               ... Repair Symbol Data ... :
+-----+-----+-----+-----+-----+-----+-----+-----+

```

Transfer Number: The numeric identifier of the Transfer that this source block is part of, encoded as a 32-bit unsigned integer in network byte order.

FEC Encoding ID: A FEC Encoding ID, as defined in Section 5.6 of [RFC6363], encoded as an 8-bit unsigned integer in network byte order.

FEC-Scheme-Specific Information elements: Zero or more FEC-Scheme-Specific Information elements as defined in Section 5.5 of [RFC6363].

Repair FEC Payload ID: The Repair FEC Payload ID as defined in Section 5.4.1 of [RFC6363].

Repair Symbol Data: The octets of the repair symbols of the Transfer, with the length calculated as the Message content length excluding the length of the Transfer Number, FEC Instance ID, and Repair FEC Payload ID.

5. Security Considerations

This new Messages and mechanisms in this document do not add additional security considerations, nor impact the existing security considerations outlined in [BTPU] and [RFC6363].

6. IANA Considerations

IANA is requested to assign new values from the "BTPU Message Types" registry for the new Message types defined in this document: TBD1, TBD2, TBD3, TBD4.

7. References

7.1. Normative References

- [BTPU] "Bundle Transfer Protocol - Unidirectional", February 2025, <<https://datatracker.ietf.org/doc/draft-ietf-dtn-btpu>>.
- [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>>.
- [RFC6363] Watson, M., Begen, A., and V. Roca, "Forward Error Correction (FEC) Framework", RFC 6363, DOI 10.17487/RFC6363, October 2011, <<https://www.rfc-editor.org/rfc/rfc6363>>.
- [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>>.

7.2. Informative References

- [RFC3095] Bormann, C., Burmeister, C., Degermark, M., Fukushima, H., Hannu, H., Jonsson, L., Hakenberg, R., Koren, T., Le, K., Liu, Z., Martensson, A., Miyazaki, A., Svanbro, K., Wiebke, T., Yoshimura, T., and H. Zheng, "RObust Header Compression (ROHC): Framework and four profiles: RTP, UDP, ESP, and uncompressed", RFC 3095, DOI 10.17487/RFC3095, July 2001, <<https://www.rfc-editor.org/rfc/rfc3095>>.

Appendix A. Acknowledgments

The author is indebted to the authors of the FECFRAME framework, and hopes that its successful application in areas outside RTP validates all the obvious hard work that went into making RFC6363 generic and reusable.

Author's Address

Rick Taylor
Aalyria Technologies
Email: rtaylor@aalyria.com