



BTPU Updates

[ietf://dtn/meetings/125](https://ietf.org/dtn/meetings/125)

[draft-ietf-dtn-btpu](#) & [draft-ietf-dtn-btpu-fec](#)

rtaylor@aalyria.com



Recap: What is BTPU?

- A Convergence Layer Protocol for BPv7
- Designed for **unidirectional, unreliable, frame-based** link layers
 - Space links (USLP, TM, AOS, DVB-S2X), also 5G, Ethernet
- Key features:
 - Segmentation of bundles across link-layer PDUs
 - Interleaving of transfers for priority (no head-of-line blocking)
 - Message repetition for loss protection (no return path needed)
 - TLV message format designed for line-rate HW/SW implementation
- No IP required

Why BTPU? Stack Reduction

Traditional (e.g. LTP over CCSDS):

BPv7

LTP

EPP

AOS / USLP

SLE

With BTPU:

BPv7

BTPU

AOS / USLP

SLE

Eliminates LTP and EPP — fewer layers to implement, less overhead, simpler to deploy, and feasible in hardware at line rate.

Change 1: Ascending Segment Indices

Before (draft-ietf-dtn-btpu-01): Segments indexed **descending** $N \rightarrow 0$

- Transfer Start carried index N (first segment, highest index)
- Final segment had index 0

After (draft-ietf-dtn-btpu-02): Segments indexed **ascending** $0 \rightarrow N$

- First segment has index 0
- Transfer End carries index N (final segment)
- More intuitive ordering
- Receiver knows transfer is complete when it has segments 0 through N

Change 2: Message Type Rename

Before (draft-ietf-dtn-btpu-01):

- Type 3: Transfer **Start** Message (first segment, signalled beginning)
- Type 4: Transfer **Segment** Message (subsequent segments)

After (draft-ietf-dtn-btpu-02):

- Type 3: Transfer **Segment** Message (any segment)
- Type 4: Transfer **End** Message (final segment, signals completion)

Completion signalling moved from first message to last. Type 3 is now a generic segment — no special “start” needed.

Change 3a: Metadata Items → Hint Items

Renamed “Metadata Items” to “Hint Items”

- Better conveys that the information is **advisory** and supplementary
- Messages (and their Hints) can be lost — hints are about the Transfer, not the Message
- Flag renamed: M (Metadata) → H (Hint)

Change 3b: Hint Item Chaining

Before (draft-ietf-dtn-btpu-01): 8-bit Metadata Type, no explicit chaining mechanism

After (draft-ietf-dtn-btpu-02): 7-bit Hint Type + **H flag** for chaining

- H=1: Another Hint Item follows
- H=0: Message content follows
- Explicit end-of-hints signalling via chained H flags
- Specs defining new Hint Types **MUST** state which Message types they apply to

Change 4: RFC 2119 Keyword Cleanup

Systematic review of lowercase “may” / “should” usage:

- “may” expressing possibility → “can” or “might”
- “may” expressing permission → “MAY” (RFC 2119)
- “should” → appropriate RFC 2119 term

Examples:

- “Messages may be lost” → “Messages **can** be lost”
- “a sender may emit duplicate Messages” → “a sender **MAY** emit”
- “may require” → “**might** require”

Change 5: Transfer Window Rationale

Added explanatory text for the half-number-space margin:

“This margin allows a receiver that has experienced a temporary loss of signal to correctly resume processing, distinguishing genuinely new Transfers from those that have wrapped around the number space and are obsolete.”

Clarifies why the algorithm uses $(2^{32} / 2) + (\text{WINDOW_SIZE} / 2)$ as the boundary for determining new vs. old Transfer numbers.

Change 6: IANA Registry Updates

BTPU Hint Types (was “Metadata Types”):

- 7-bit type space (was 8-bit) due to H flag in header
- 0..0x6F Standards Action (was 0..0xBF)
- 0x70..0x7F Private Use (was 0xC0..0xFF)
- New requirement: specs MUST state applicable Message types

BTPU Message Types — Type 3/4 renamed

BTPU Message Flags — Bit 0: H (was M)

Other Minor Changes

- **AOS reference URL fixed** – was incorrectly pointing to TM document
- **YAML metadata cleanup** – removed unnecessary quoting
- **Grammar fixes** – “a Indefinite” → “an Indefinite”, etc.
- **Receiver duplicate handling** – added explicit text: “A receiver MAY ignore any duplicate Message already received for an in-progress Transfer”

Open Issues

- Transfer Window size: recommended value of 16 – is this appropriate?
 - “16 is an arbitrary number, and needs discussing by the WG”
- Transfer Window size range: at least 4, less than 2^{12}
- Any other feedback on the Hint Items mechanism?

BTPU-FEC Update (draft-ietf-dtn-btpu-fec-01)

Companion draft: FEC extension for BTPU using the FECFRAME framework (RFC 6363)

Changes since draft-ietf-dtn-btpu-fec-00:

- **Terminology aligned with RFC 6363** – “Source Block” replaced with “Application Data Unit” (ADU), clarifying distinction from RFC 9171 ADUs
- **New FEC Transfer Operation section** – completion semantics (no Transfer End; FEC scheme determines completeness), message mixing rules, Transfer Cancel usage
- **Expanded Security Considerations** – FEC provides no authentication; receivers SHOULD use BPSec for integrity verification of reconstructed bundles
- **IANA table added** – four new message types (Pre-agreed/Explicit Source/Repair)
- **BCP 14 and grammar fixes** throughout

Next Steps

- Solicit WG feedback on open issues
 - Further editorial review
 - Implementation experience?
-

Thank you
