Bundle Transfer Protocol - Unidirectional (BTPU)

DTN Working Group, IETF-122 Bangkok

Rick Taylor, Aalyria

Applicability

In the realm of near- and deep-space communication there are a number of standardized link-layer protocols:

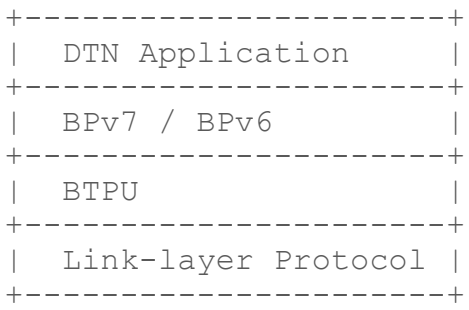
- USLP
- EPP/SPP
- TM
- AOS
- DVB-S2X
- 5G
- Ethernet
- … and more

They all share some common characteristics:

- Frame-based
- Unidirectional (no in-band return path)
- Unreliable
- Point-to-multipoint

Exploiting commonality

BTPU is a Convergence Layer Protocol designed to transport BPv7 Bundles, sufficiently general to be applicable over all such link-layer protocols.



It is however generic enough to carry any blob, rather than just Bundles.

Mechanisms

The protocol builds on 3 well-known concepts:

1. Simple Type-Length-Value binary encoding of Messages, to allow performant frame-packing and processing.
2. Segmentation of bundles into parts to fit into the underlying frame.
3. Sliding window to allow multiple bundles Transfers to be in-progress at a time.

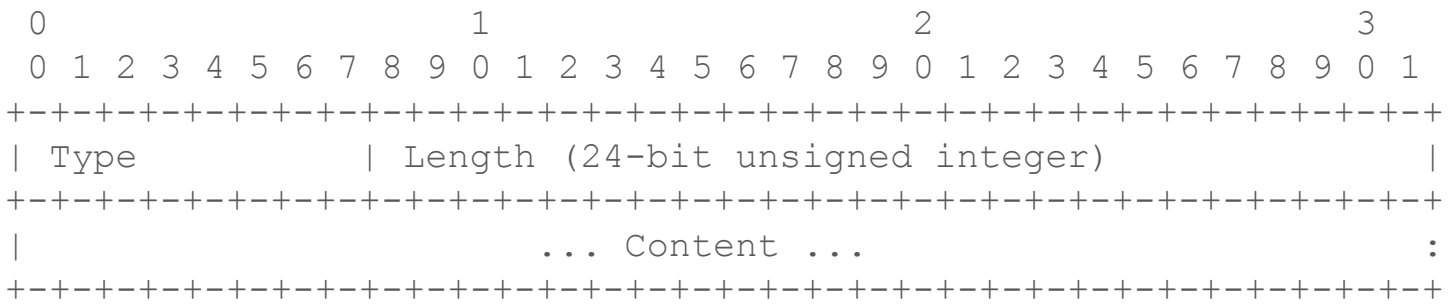
To prevent against data loss, two optional mechanisms are provided:

1. Explicit Message repetition.
2. Forward error correction coding, as a second draft.

Messages

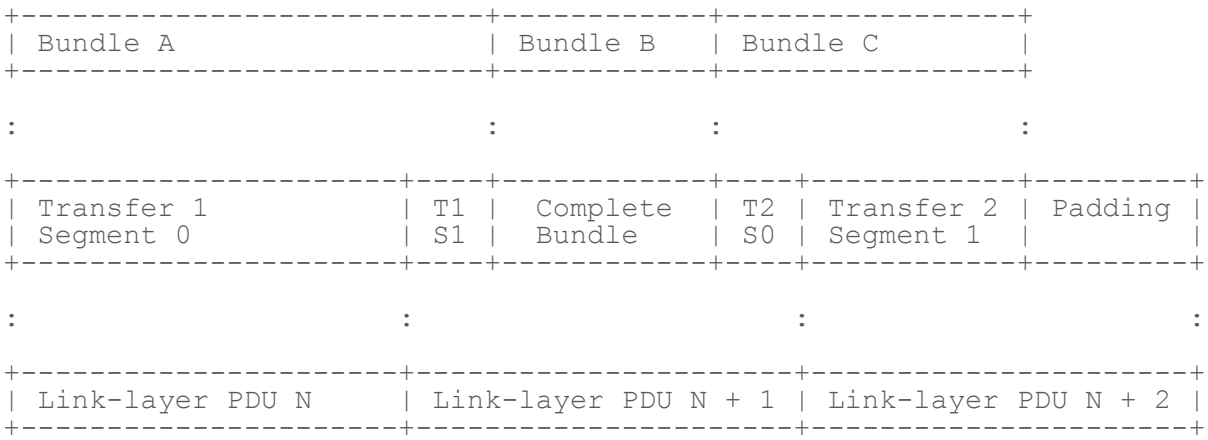
Everything in BTPU is a Message, including padding. Messages are the basic unit that is packed into a link-layer PDU.

Messages have a common header:



Segmentation

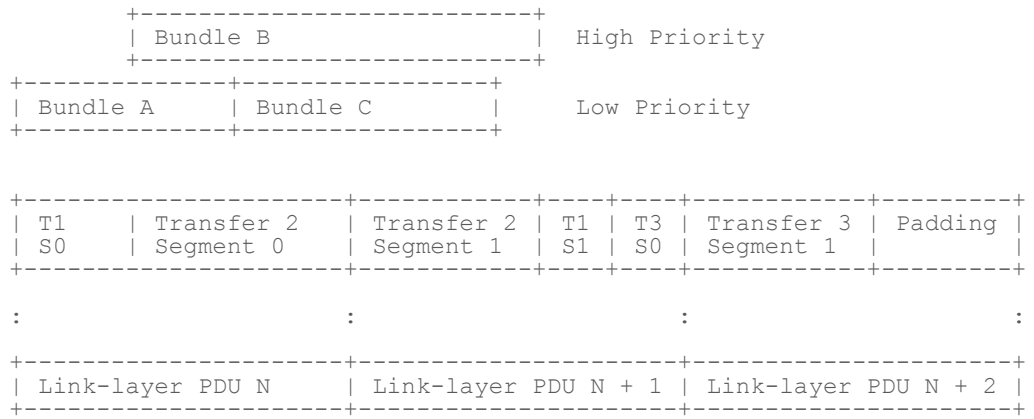
Because a Bundle is of variable length, which is unlikely to be exactly the same size as a Link-layer PDU, Bundles are divided into Segments as required, such that each Link-layer PDU can be efficiently filled, and Bundles can be transferred in a minimal number of Link-layer PDUs.



Transfer Multiplexing

Individual Messages of one Transfer may be interrupted by Messages of another Transfer, and Transfers may be explicitly cancelled while in progress.

This avoids head-of-line blocking, and provides a primitive for some kind of transfer prioritisation.



Sliding Window

Because there is no in-band return path, there must be a synchronisation mechanism to allow receiver to detect if a received Message is out of date, and when it is no longer worth waiting for a missed Message.

- Every Bundle has a 32-bit unsigned integer Transfer ID, that is allowed to roll over.
- The sender and receivers maintains a window of preconfigured size, and only Transfer IDs in the window are in progress.
- When a new Transfer ID ahead of the window is received, the window is adjusted forwards through the number space, and old Transfers no longer in the window are cancelled.

Message Repetition

A sender may repeat any Message if:

- It is repeated without alteration.
- It is within the sliding window.

This provides a basic level of protection against data loss, but does allow some unusual usages:

- Some Transfers may have a different amount of Message repetition, providing different levels of reliability.
- A set of Transfers may be repeated for an extended period, providing a ‘data fountain’ capability.

Forward Error Correction

The second draft (https://datatracker.ietf.org/doc/draft-taylor-dtn-btpu-fec/) provides an entirely optional mechanism for using FEC on a per-Transfer basis.

Its is built on the pre-existing FECFRAME framework, RFC6363, developed for RTP, but applicable to BTPU.

It provides the following FEC schemes, and is extensible to new emerging schemes:

- Raptor and RaptorQ
- LDPC-Staircase
- Reed-Solomon
- Sliding Window Random Linear Codes

Coming soon...

- BTPU-over-Ethernet
- BTPU-over-DVB(S2X) (with IEEE/ETSI)
- BTPU-over-5G PDU (with 3GPP)
- BTPU-over-CSSDS formats (with CCSDS assistance)

Any questions?