Meeting Mission Needs with Delay/Disruption Tolerant Networking (DTN)

Jonathan Wilmot (NASA/GSFC)
JP Swinski (NASA/GSFC)

Delay/Disruption Tolerant Networking (DTN) has the potential to significantly improve current mission data transfer capabilities and add the network layer multi-hop data transfer capability required for future missions. Missions have been reluctant to adopt DTN as they do not see existing DTN implementations as having supported "real" mission use cases and meeting requirements for operability, performance, resource utilization, management, and fault detection/recovery.

For mission infusion to occur, a DTN implementation must show that mission use cases can be met and that the overall cost and/or risk is, or can be, lower than existing approaches. Currently, PACE is the only NASA Category 2 or higher mission where DTN is to be integrated as a part of normal operations with the mission requirements as listed above. It is important that PACE be a success case as an early adopter and not an example of "let's not do that again"

This document has three main sections. The first, discusses capabilities missions currently have, the second provides a typical PACE-like mission context, and the third lists requirements on DTN to achieve the capabilities from the first section.

1. Current mission capabilities:

- Data sequence and integrity tracking at each layer of the network stack.

  These features are used for data reconstruction, and fault detection/recovery. Knowing which functional block is generating faults is key to fault isolation and recovery. This is especially important during development were subsystems implementations may have bugs and edge case faults dropping PDUs at different and sometimes multiple layers.

- Data flow channelization, prioritization, bandwidth partitioning, and storage partitioning.

  Starting at the spacecraft end node, data from different sources is allocated a bounded space in buffer and storage memory and subsequently allocated bandwidth at time of transmission based on system mode. For example, science data from instrument one is allocated a 1 MiB PDU input buffer, and then 200GiB of storage, and 100Mbps when transmitting in nominal science mode with a station contact rate of 120Mbps. For a station contact of 80Mbps, instrument one is allocated 70Mbps. For an S-band station contact of 2Mbps, instrument one has zero allocated bandwidth. Note that all of these allocations are modifiable via command from the mission control center or on-board automation.

  Flows are designated in CCSDS AOS by virtual channel and "routed" based on channel. For example: Instrument one data in VC1. Spacecraft real-time data in VC0.

- Storage optimization, recovery and management.

  As storage partition(s) fill, policies are implemented to allow data overwrite based on oldest or to halt all new data stores.

  Where possible, data is only stored once and then "pointed to" for subsequent data transfers. (memory copy or DMA)

  Data sets and partitions can be deleted upon command.

- No data loss due to errors in time management.

  Data is not discarded due to invalid time stamps. Although some analysis is required, data sequence and time can be reconstructed based on data within the packets and external meta-data. This information is critical to anomaly investigation and subsequent system recovery and fault masking.

- Data can be retransmitted or dropped as configured.

  Data from different flows can be held and retransmitted as configured. Once the data unit has been confirmed as received the sender can delete.

  Requests for retransmission can take a different path and have different latency and bandwidth allocations.

- Multi-hop data relay with different bandwidth and quality of service between hops.

  For LEO spacecraft the typical scenario is high speed to ground station with a lower speed transmission from the ground system to the control or science center.

  For deep space missions the typical scenario is resource constrained rover using reliable store and forward protocol to transmit to orbiter subsequent long-haul retransmission.

- Data link abstraction and use optimization.

  Protocol data units (PDU) are able to span certain PDUs at lower layers. Although this currently cannot occur at every layer there is a desire to do so. Fragmentation and the CCSDS M-PDU are example mechanisms to accomplish this.

The above capabilities have been implemented in many different ways at significant costs to each mission. DTN is envisioned to provide a common implementation that can meet these use cases.

2. Spacecraft Mission Context

The context for telemetry egress over DTN on an example spacecraft similar to PACE is shown in Figure 1 Telemetry Egress Context Diagram. Within the spacecraft, there are several application data sources that are agnostic to, and abstracted from the underlying mechanisms to exchange data locally and/or externally. These applications generate application protocol data units (APDU) that are sent via a message transfer service to one or more applications that have subscribed. A telemetry service application receives these APDUs as configured to transmit to external systems. A data storage service application also receives APDUs as configured to store the APDUs for later downlink and/or additional onboard data processing. DTN services are implemented below telemetry services and are considered just another protocol stack for telemetry egress. The configuration of the data stores and telemetry services are all controlled from command APDUs both from onboard automation and/or commands from the operations center(s).

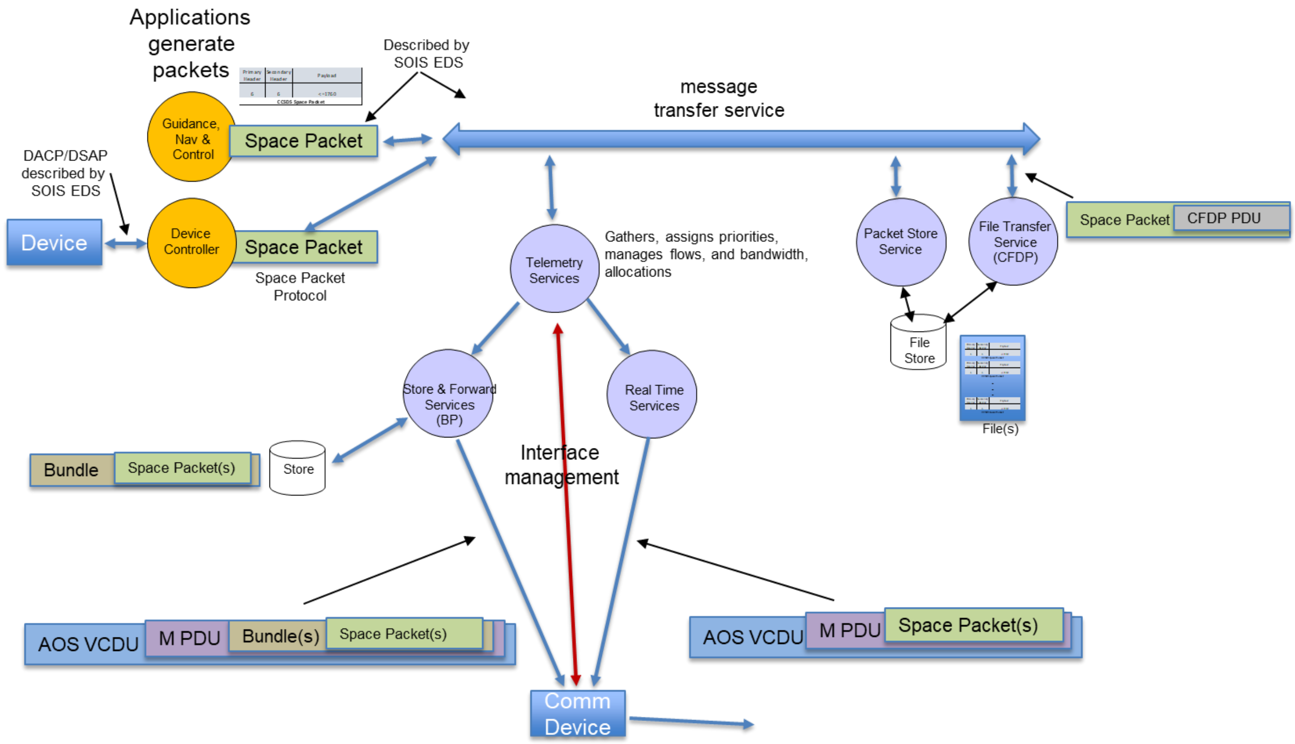

Figure 1 Telemetry Egress Context Diagram

The context for APDU uplink over DTN on an example spacecraft similar to PACE is shown in Figure 1 Telemetry Egress Context Diagram. Within the spacecraft, applications subscribe to APDU data sources based on topic. One of the possible data sources is the external uplink (ingress) protocol stack. Currently that stack utilizes CCSDS framing and CCSDS COP-1 for reliable and in-order delivery of APDUs. On-board applications assume this in-order delivery. To avoid changes to on-board applications and services DTN would need to support in-order delivery as well.

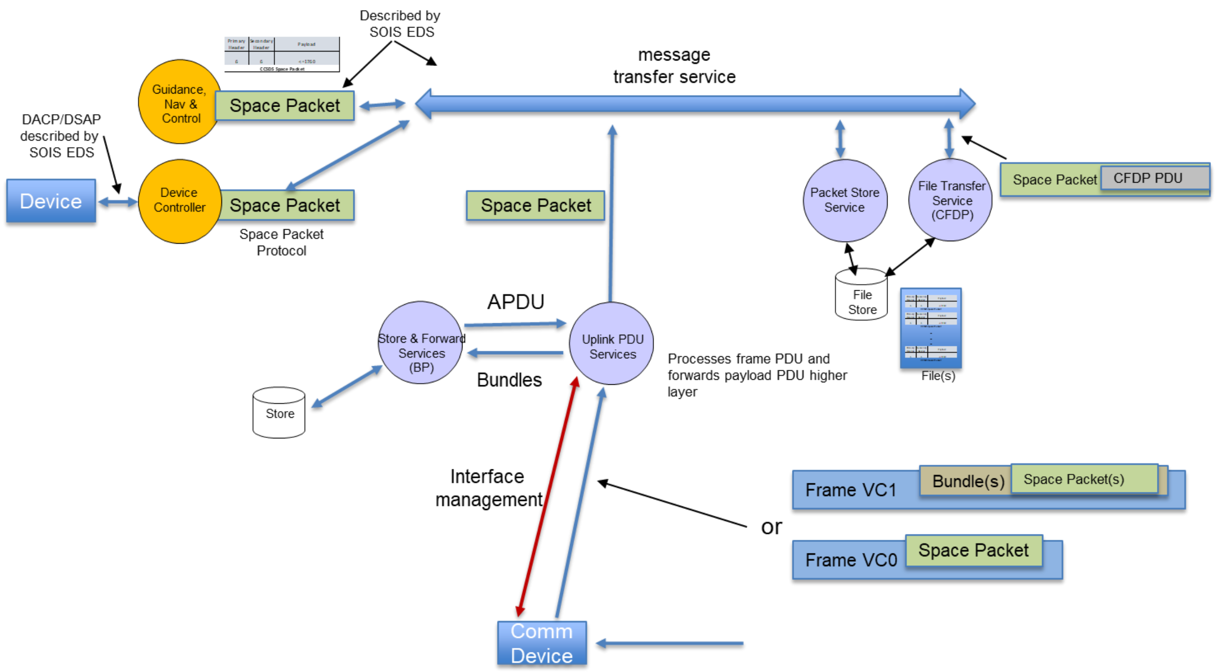

Figure 2 APDU Ingress Context Diagram

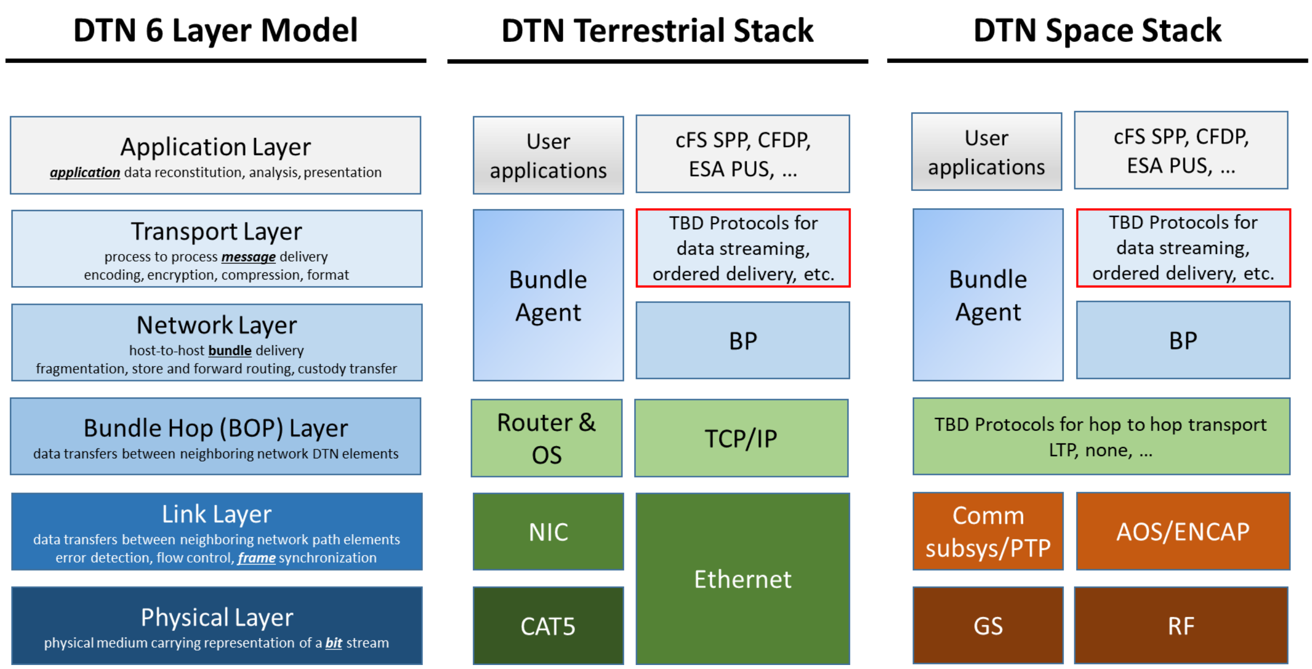

Figure 3- Simplified DTN Stack

3. Mission Requirements

As missions start to adopt DTN they are looking at impacts to mission operations, data flows, performance, resource utilization, fault detection and recovery. Mission use cases require clarifications, additions, or changes to CCSDS BP Blue book. The goal is to have standards based mission implementations and not point solutions for each mission. The following requirements are based on the current mission capabilities listed above.

1. **Requirement:** DTN shall support the unambiguous determination of bundle transmission sequence.

   Rationale for change: Support for data sequence tracking at each layer of the network stack. The currently defined creation timestamp and sequence are insufficient to establish the original sequence of the data sent by the spacecraft (end node).

   - Since the sequence number can be reset at least every second, there is no way to determine if there are gaps in the data.
   - Since the creation timestamp is subject to the spacecraft's timekeeping system, the time could go backwards or have jumps (either as a result of a time correction, a transition to another onboard time reference, or as a part of testing or manual operations).
   - Ground data processing will not be able to determine if there are gaps in the data or if the data has been sent out of order.
   - Knowledge of performance and errors at each layer is important for network analysis and fault diagnosis

   Proposal: Add recommended practice that the Creation Sequence Number is a monotonically incrementing integer that will roll to 0 once a MIB defined maximum value is generated. (PACE is using a 28(?) bit number stored as a UINT32 SDNV)) The sequence number is independent of the creation time. Allowing this practice at the Bundle layer will allow early detection and accounting of gaps (even in the presence of encryption) Use of sequence number in transport above BP should be discussed as well.

2. **Requirement:** DTN shall have robust and configurable behavior (minimal/zero data loss) due to loss of time faults in network.

   Rationale for change: Nodes can lose time reference due to transient system faults. (Reset, loss of GPS, …) Current systems do not lose any data as a result of bad time information in the PDU headers. Currently BP would drop or time out bundles with bad time at what is likely a critical event in the mission. Note that time can be in the future or past and the fault may persist for extended periods until local time can be corrected. (ISS issue was future time)

   Proposal: Revisit DTN Bundle Age Block for Expiration without UTC, draft-irtf-dtnrg-bundle-age-block-01 and include in CCSDS Blue Book. Another approach would be to have a flag associated with the time indicating confidence. The policy on how to deal with the issue is TBD. Create MIB parameter in BP ADM to configure.

3. **Requirement:** DTN shall support management and partitioning of multiple traffic flows.

   Rationale for change: Spacecraft resources are managed and allocated based on "flows". A flow may be a data source, data sink, data type, quality of service attribute, or any combination of those. Flow management is required at each "BOP" Figure 3- Simplified DTN Stack.

   Proposal: Define flows within the DTN specification and create an Application Data Model (ADM) for real-time command and telemetry management of those flows. The parameters needed to manage a flow shall be in unencrypted header blocks (TBR). The header block may include or be included in integrity parameters. Register flow ADM parameters in SANA or include directly in Blue book?

4. **Requirement:** A required minimal set of operational command and telemetry management interfaces shall be specified.

   Rationale for change: Many spacecraft during operations and in development use command and telemetry interfaces based on CCSDS standards. These interfaces are used for both ground control and onboard automation for operations and fault management. To ease infusion into upcoming missions, DTN should provide a standard service management interface that can be utilized by existing operational and on-board automation services. This would provide an ability to easily integrate DTN configuration with existing power, attitude control, command and data handling configuration services. A well define management interface could also be used by future automation engines (such as AMP) without changing the underlying service interface.

   Note, the interfaces should be split between those required for real-time operations and those that may be part of a more static MIB.

   Proposal: Create a DTN Application Data Model (ADM) for real-time command and telemetry interfaces. cFS team will auto generate CCSDS SEDS from the json ADM to create architecture specific bindings for the required management interfaces. ESA, KARI, ISS, and CCSDS MOIMS and others can create architecture specific bindings based off the standard ADM as well without impacts to their existing architectures. Register ADM in SANA or include directly in Blue book?

5. **Requirement:** DTN Aggregate Custody Signal (DACS) behavior shall be fully specified.

   Rationale for change: The CCSDS blue book does not define interoperable behavior for sequence number rollover and other DACS behaviors. PACE has requested that DACS be unique to flows to allow deterministic partitioning of node processing, storage, and bandwidth resources. (Similar to use of CCSDS Virtual Channels now)

Proposal: Define DACS flow behavior and DTN MIB configuration parameter for maximum DACS. This parameter may be defined per "BOP" and be populated in the MIB by the originating node via a configuration bundle. Define custody signaling as a per flow mechanism to support DACS/flow pairing. (Also supports management requirement 3)

6. **Requirement:** Bundles shall support a separate integrity block for payload data.

   Rationale for change: The integrity block is intended to be an end to end check that the data has not been corrupted in transmission or in node storage. The block may need to present along with other security blocks. Analysis is needed to determine DTN behavior when an integrity or security block fails. i.e retransmission (DACS) as though it failed in transmission or some other behavior.

   Proposal: Define a standard integrity block with self-identifying block check algorithms registered in CCSDS SANA.

7. **Requirement:** DTN shall specify a selectable transport service above the BOP layer to provide delivery of application PDUs in end node transmission order.

   Rationale for change: The current Delay Tolerant Payload Conditioning (DTPC) specification is incomplete and does not support selectable transport attributes. Some end user applications require in-order delivery, others require duplicate suppression, and some require both.

   Proposal: Specify a selectable transport service above the BOP layer to provide delivery of application PDUs in end node transmission order. Include the concept of a sliding window and edge case analysis. As an implementation optimization it is strongly suggested that the transport layer to BOP layer Service Access Point (SAP) provide a mechanism for the BOP layer to retain the bundle until the transport has explicitly acknowledged it. It is assumed that ION has such a mechanism (zero copy object) in the DTPC to BP service interface.

8. **Requirement:** DTN shall specify a service to provide duplicate suppression.

   Rationale for change: The current Delay Tolerant Payload Conditioning (DTPC) specification is incomplete and does not support selectable transport attributes. Some end user applications require in order delivery, others require duplicate suppression, some require both.

   Proposal: Specify service to provide duplicate suppression. Note that this can be handled in the bundle layer (with requirement #1), or above the bundle layer in the transport layer.

Suppression in the bundle layer has benefits at each hop as storage and processing resources are not require to store and forward the duplicate.

It is important to note that some of the requirements above are included in upcoming CCSDS or IETF standards. It would be in the interest of all that a forward path be established such that the PACE instantiation is not a point solution and adopts those forward looking solutions. An example of this is DACS is not present in IETF DTN version 7 and has been replaced by a bundle in bundle approach. This can be viewed by projects as an example that **DTN is in flux and is not ready for deployment!**