App-Layer Multiplexing in NOS3 + cFS

Notes for System Integration

Joshua Wille

Overview

In a NOS3 + cFS setup, the role typically called the "application-layer multiplexer" is realized by the **cFE Software Bus (SB)** together with a **Telemetry Output** application that drains selected messages to ground. Flight apps (IMU, GPS, HK, payload, etc.) publish *CCSDS telemetry packets* (with APIDs/MsgIDs) onto the SB. A dedicated output task—lab-simple `to_lab` or a mission `to` app—*subscribes* to a configured set of MsgIDs and *forwards* them over UDP to the ground tool (COSMOS). Operationally, "the mux" is therefore the combination of: (i) which producers publish at what cadence, (ii) what the output app subscribes to and filters, and (iii) how the output app paces/queues outbound packets.

Key Components and Responsibilities

cFE Software Bus (SB). Provides publish/subscribe messaging among cFS apps. Each message already carries a CCSDS primary header (APID/MsgID, sequence, length). The SB enforces message typing and routing but does not itself impose QoS beyond buffering.

Scheduler (SCH). Triggers producers at defined rates (e.g., IMU at 50 Hz, GPS at 1 Hz, HK at 1 Hz). These rates strongly shape the interleaving that the output app sees because they determine when packets arrive to be forwarded.

Telemetry Output app. In NOS3 labs the default is `to_lab` (a simple, non-flight utility). It subscribes to a table-defined list of MsgIDs and immediately forwards received packets via UDP to the ground. Missions often swap in a more configurable `to` app that supports multiple routes/channels (e.g., separate sockets or physical links) and basic filtering/selection. Together with SCH cadence, this app is the practical "multiplexer."

Command Ingest (CI). Symmetric path for uplink: receives CCSDS commands from ground over UDP, validates headers, republishes them on SB. CI is the demultiplexer counterpart for commands.

Scheduling & Policy (What Makes It a "Mux")

Inclusion & order. Inclusion is decided by the Telemetry Output subscription/filter table; order emerges from arrival times at the output app, which are governed by producer rates (SCH) and any output-app internal queuing/pacing. A lab build with `to_lab` behaves essentially "as fast as packets arrive," while a mission `to` may group, route, or rate-limit by channel.

Per-flow shaping. In lab form there is no sophisticated per-APID QoS; test systems rely on SCH to avoid floods. In mission form, the `to` app can segregate flows by "route/channel" and apply simple pacing so high-rate payload does not crowd out housekeeping or C&C.

End-to-End Framing Context (Mapping to CCSDS)

At the packet level, every SB telemetry message is already a CCSDS *Space Packet* (SPP): it has APID/MsgID, sequence flags/count, and length per CCSDS 133.0-B-2. The output app emits those SPP packets over UDP to the ground tool. If/when a true space data link (TM/AOS/TC) is introduced beneath, those same interleaved SPP bytes are packed into transfer frames; TM uses the *First Header Pointer (FHP)* to indicate where the first new packet header begins in each frame, AOS uses *M_PDUs* within virtual channels, and TC uses its own segment header. In NOS3 labs, that lower layer is abstracted as UDP, but the app-layer multiplexing model aligns with flight practice.

What You Configure in NOS3

- **SCH tables:** set producer cadences (e.g., IMU 50 Hz, GPS 1 Hz).
- **Telemetry Output subscription table:** which MsgIDs/APIDs to forward, optional filtering/rate hints (mission `to` only).
- **UDP bindings:** destination IP/port for COSMOS; matching interface in ground config.

Traceable References (quick links)

- cFS training overview of SB, CI/TO roles: NASA cFS Training (NTRS).
- NOS3 documentation (cFS scenario, CI/TO notes): NOS3 Scenario – cFS.
- Telemetry Output Lab app (`to_lab`) description: cFS Catalog; source: GitHub: nasa/to_lab.
- CCSDS Space Packet Protocol (APID, sequence, length): CCSDS 133.0-B-2.
- CCSDS TM Space Data Link Protocol (FHP semantics): CCSDS 132.0-B-3.