Checklist for Full CCSDS / cFS Comms Stack (Application, Data-Link, Physical Layers)

Your Team Name

October 24, 2025

Introduction

This document captures the full checklist for implementing a communications stack for a small-sat using the *Open Systems Interconnection (OSI)* model, via *CCSDS* standards and *cFS* flight software. The stack covers the Application Layer (L7), Data-Link Layer (L2), and Physical Layer (L1), and the "thin waist" interface between PetaLinux/FPGA and flight software. It assumes the GRC (GNU Radio Companion) system already covers the core PHY and Data-Link processing, so the remaining items represent all other necessary artifacts, software applications, and operational deliverables.

Application Layer (OSI Layer 7): CCSDS on cFS (plus CFDP if used)

a. **Define your L7 services and PDUs**

- Space Packet schemas for telemetry (HK, ADCS, payload, link health): fields, units, validity flags, sequence counters, producer timestamps.
- Command dictionary: opcodes, arguments, valid ranges, failure codes; ACK/NAK policy and confirmation semantics.
- CFDP profile (if used): class-1 (unreliable) vs class-2 (reliable), segment sizes, transaction IDs, timers (ACK, NAK, inactivity), custody transfer policy, filestore directives, condition codes (see CCSDS 727.0-B-5). [1]

b. **IDs, routing, and QoS**

- APID allocation table (unique per service).
- APID → VCID mapping and priority classes (e.g., Commands ¿ HK ¿ Payload).
- Target rates per APID (packets per second / bits per second) and maximum bursts for each.

c. **cFS edge apps (the only flight-software you must write/touch)**

- COMMS_TX: subscribes to SB, timestamp, queue by VC, token-bucket shaping, serialize to driver (e.g., `/dev/commstx`), expose link stats (queue depths, drops, achieved bps per VC).

- COMMS_RX: read records from driver (`/dev/commsrx`), validate Space Packet primary header, publish on SB, expose RX stats (frame counts, CRC/RS fails, ASM resyncs).
- (Optional) CF app: configure if using CFDP; ensure CFDP PDUs ingress/egress via your CLA shim.

d. **Ground peer**

- Ground command encoder/telemetry decomm tool that understands your APIDs command dictionary.
- Ground CFDP engine (if used): storage path, checksums, transaction dashboard.
- Convergence Layer Adapter (CLA) shim on both ends (flight ground): raw CFDP PDU in/out via socket, length-prefixed handshake.

e. **Time versioning**

- Use `cFE_TIME` (PPS/GPS disciplined) for producer timestamps and uptime.
- Protocol version fields, optional TLVs, and capability negotiation logic.

f. **Security (if in scope)**

- Define where applied (L7 vs L2): MIC/auth tag format, replay window, key IDs, and failure behavior.

CCSDS Data-Link Layer (OSI Layer 2): Framing, Coding, Managed Parameters

a. **Transfer frame flavor layout**

- Choose TM/TC or AOS; fixed frame length (bytes) and Transfer Frame Data Field (TFDF) policy (packetized vs M_PDU segmentation).
- Define FHP (First Header Pointer) rules, idle-packet insertion, and usage of VCID / MCID.
- If needed: OCF/CLCW usage for uplink loop control.

b. **Randomization sync**

- ASM (Attached Synchronization Marker) pattern (e.g., 0x1ACFFC1D), insertion/detection policy; sync search, slip counters, reacquisition thresholds.
- Pseudo-randomizer: enable/disable, seed, polynomial per CCSDS standard.

c. **Error-control coding**

- Legacy: RS(255,223), interleave depth I; convolutional K=7, rate 1/2 (and other punctured rates if used).
- Modern: LDPC or Turbo codes as defined in newer CCSDS standards.
- CADU composition: ASM + coded frame (+ RS parity) + optional idle fill, correct bit/byte ordering.

d. **VC multiplexing admission**

- Per-VC token buckets (rate R_{vc}, burst B_{vc}), and weighted fair queuing or strict priority across VCIDs.
- "Link up" gating (TDMA windows or antenna pointing), scheduler drains only when enabled.

e. **Managed parameters (link config)**

- SCID, VCIDs, frame length, coding mode, interleave depth, randomizer on/off, ASM select.
- Telemetry counters: frames sent/received, frame quality, RS/LDPC fails, ASM slips, throughput per VC.

f. **RX deframing behavior**

- Resync strategy when ASM is lost, handling of bad frames (drop vs deliver with flags), tagging delivered Space Packets with quality indicators.

Physical Layer (OSI Layer 1): Modem, Impairments, RF Hooks

a. **Waveform**

- Modulation scheme: BPSK, QPSK, OQPSK; symbol rate, RRC roll-off α, filter span.
- Symbol mapping (Gray), pulse shaping, gain staging.

b. **Acquisition tracking**

- Preamble/pilot (if applicable), timing recovery loop bandwidth (PFB or Gardner), carrier loop (Costas) bandwidth, AGC policy.
- Doppler tolerance, pre-comp plan derived from ADCS/ephemeris.

c. **Channel model scheduling**

- AWGN plus frequency offset; optional fading or multipath if needed.
- TDMA or scheduled link windows; quiet periods during slews or pointing manoeuvres.

d. **RF plan (external but required)**

- Frequency allocation, Tx power / EIRP, spectral mask, antenna patterns/pointing windows, regulatory compliance.

PetaLinux / Driver / FPGA Seam (The "Thin Waist")

a. **Device interface (flight CPU FPGA)**

- Device nodes: `/dev/commstx` and `/dev/commsrx`.
- Record header format: `LEN(2B) | FLAGS(1B) | VCID(1B) | TIMETAG(6B) | SPACE_PACKET(LEN).`
- IOCTLs: set/get coding mode, symbol rate, VC enable mask, randomizer on/off, ASM select; read hardware stats.

- DMA/AXI config: buffer size/alignment, interrupts/credits, non-blocking E_BUSY behavior.

b. **FPGA expectations**

- Treat Space Packets as opaque: no APID parsing in PL logic.
- TX path: pack → randomize → FEC encode → modulate; RX path: demodulate → FEC decode → derandomize → deframe → output Space Packets.
- (Optional) Per-VC token bucket support in PL if you enforce shaping in hardware.

Ground Segment (Closing the Loop)

a. **Decode monitor**

- Ground deframer/decoder matching the flight L2/PHY choices for bench or operational use.
- Ground CFDP engine (if used): supports filestore, transaction logging, checksums, dashboard.

b. **Command control**

- Command encoder respecting your dictionary, safety interlocks, APID map.
- Telemetry decomm: APID map → channels → plots; logging, archiving with timestamps aligned.

Validation & Operations Artifacts

- Bit-exact vectors: Space Packet → Frame → CADU (TX) and back (RX).
- BER/FER vs. E_b/N_0 curves per modcod; ASM resync statistics vs frequency offset.
- Queue drop logs per VC under stress; achieved bps vs token budgets.
- CFDP transaction logs: class-1/2 transfers, induced loss/dup/reorder; selective NAK lists, retries, finished codes.
- Managed-parameter tables: SCID, VCID, frame size, coding mode.
- Runbooks: rate changes, window schedules, anomaly response.
- ICDs:
  - BYTESTREAM (CPUFPGA) header spec.
  - APIDVCID/QoS mapping.
  - L7 Spec: telemetry schemas, command dictionary, CFDP profile.
  - Ground CLA/GRC wiring: port numbers, transport records, shim spec.

References

References

[1] CCSDS 727.0-B-5, *"File Delivery Protocol (CFDP)"*, Blu eBook, Consultative Committee for Space Data Systems (CCSDS). Available at `https://public.ccsds.org/`.

[2] NASA, *"Core Flight System (cFS)"* GitHub Repository. Available at `https://github.com/nasa/cFS`.

[3] NOS3, *"NOS3 Flight Software Integration"* Documentation. Available at `https://nos3.readthedocs.io/en/latest/NOS3_Flight_Software.html`.

[4] Yamcs, *"CFDP Service for Ground Segment"* Documentation. Available at `https://docs.yamcs.org/yamcs-server-manual/services/instance/cfdp/`.