Protocol

Scheme of the CubeSat and GS chain of communication

Communication with a ground station on Earth is made possible through the use of a 2.2 GHz communication subsystem.

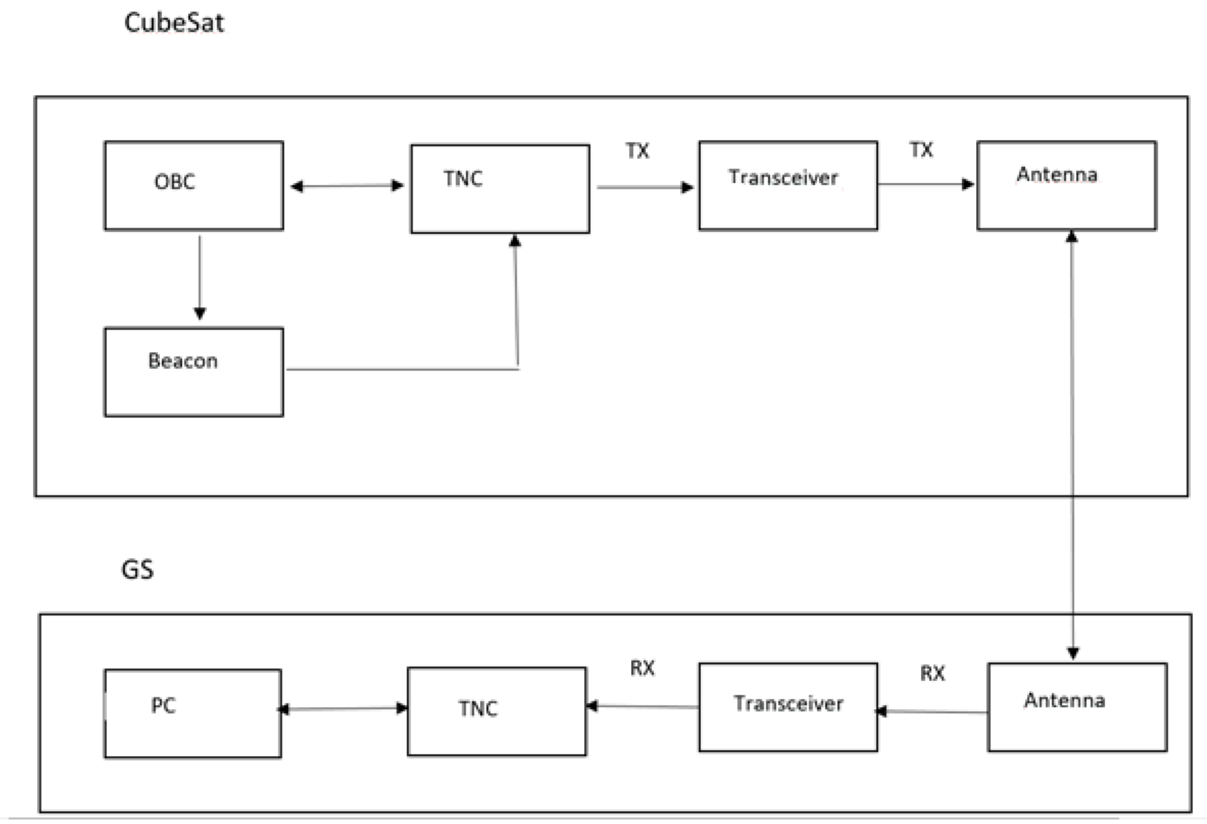

Figure 1: Block Diagram Communication

Terminal Node controller (TNC)

A Terminal Node Controller (TNC) is a device used by radio amateurs to connect to AX.25 radio packet networks. This device takes over the functions of the packet assemblies / disassemblers of the X.25 networks, and adds a modem for the conversion between digital signals and audio signals.

TX: transmit
RX: Receive
PC: personal computer

Transmission protocols

The communication subsystem of a CubeSat has two main purposes:

• Transmit telemetry data, including a beacon.

Of course, every CubeSat needs a Telemetry. We can select two simple solutions to this Telemetry requirement using the ubiquitous AX.25 amateur radio data standard. AX.25 can be received by hundreds of thousands of amateur radio operators worldwide.

The MIM Module: The simplest Telemetry module is the MIM module shown below which provides for multiple periodic AX.25 packets at 1200 baud (AFSK=Audio FSK).

1200 Baud TNC: The second and much more capable telemetry is the use of a Kantronics KPC-3PLUS TNC carved down to fit within the four-inch cube. As can be seen in the photo below, the board will fit diagonally if the connectors are removed and the power supply portion of the board are relocated. This gives the same Telemetry, Beacon, GPS and CW ID capability as with the MIM module, but includes a full TNC DIGIPEATER and 4 channel COMMAND/CONTROL channel as well. With the full TNC, a complete digital communications transponder mission, such as PCSAT, can be supported.

• Provide a mean for the satellite to communicate with ground station and vice versa. (The Ground station will receive data from the CubeSat only in this direction)

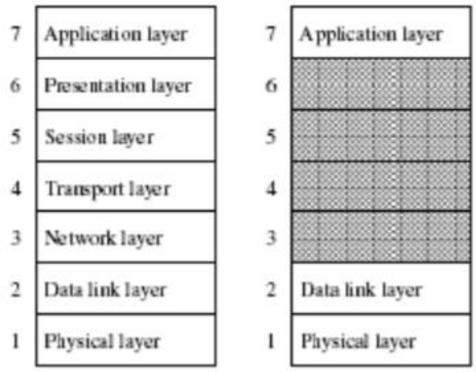

Figure 2.21: Standard OSI Model and modified Model for a CubeSat

The physical layer performs the task of transmitting raw bits over the communication channel. The data link layer's task is to take this stream of bits and transform it into a form that seems uninterrupted and without errors from the application layer. This is done by breaking the input data from the application layer into frames.

These frames are transferred to the data link layer at the other end of the connection where the data is assembled and sent to the application layer. The data link layer checks for errors in frames and missing frames and takes appropriate actions to correct this. The data link layer looks after flow control *to prevent a fast sending source from drowning a slow receiving sink.*

AX.25 Protocol

The most used data link layer protocol between radio amateurs is AX.25, and it is also well documented. AX.25 is also implemented in most commercial TNCs (Terminal node controller).

We will look at the AX.25 protocol below. The AX.25 protocol gives a complete overview of the structure of the data link layer and the interfaces between the data link layer and the layer above and the physical layer. Same of the properties with the protocol is listed below. Properties of AX.25 are:

- confirms to HDLC (High-level Data Link Control)
- supports amateur call names
- supports connected links
- supports connection less links, needed for beacon
- supports half / full duplex
- supports error detection.

The structure of the lowest layers of the communication system will follow easily from the simplified OSI Model with only three layers, see Figure 2.21. AX.25 corresponds to the data link layer. The layer above the data link layer will depend how the OBC will be implemented.

AX.25 Frames

The data units of the data link layer are called frames. There are three different types of frames:

a) Information frame (I frame)
b) Supervisory frame (S frame)
c) Unnumbered frame (U frame)

The I frames carries the data that are to be transmitted. The S frames take care of acknowledging and requests for retransmission of lost or corrupted data. The U frames are responsible for establishing and terminating link connections. With the U frames it is also possible to transmit data outside the normal flow. The AX.25 protocol envelops the data it sends to the physical layer in a frame. The structure of such a frame is seen in Figure 2.22.

Flag	Address	Control	PID	Info	FCS	Flag
01111110	112/224 bits	8/16 bits	8 bits	N*8 bits	16 bits	01111110

Figure 2.22: Frame Construction

Limitations of AX.25

A satellite will not, as a stationary radio station, be within range all the time. In fact, the satellite will only be able to communicate with the ground station for a limited duration. If the amount of data, the satellite needs to transmit is larger than the amount it can transmit during one pass it needs to transmit it over several passes. Even if the amount of data is small it is possible that the satellite needs to retransmit the last frames which were not acknowledged during the last pass, if such frames exist. The AX.25 protocol discards the buffered data when a connection is lost. To solve this problem, we need a system which keeps track of which data are transferred and can be erased from memory, and which data it need to keep in memory until next time the satellite can transmit. This system must be implemented in a higher layer.

A property that AX.25 lacks is the ability to handle priorities. All data is transmitted in the order it is sent to the TNC. Prioritization between different data sources in the satellite must also be implemented in the OBC. It is possible to send UI-frames and thereby avoiding the normal flow control, but this is a very unreliable procedure. The AX.25 protocol supports connection oriented frames with I-frames, and connectionless frames with UI-frames. The connectionless mode can be used for the satellites beacon since this form of communication is one way only and need no acknowledge responses. The connection oriented mode will be used when a ground station asks for a connection.

CCSDS Protocol

The most used transmission protocol for professional satellite telemetry is the CCSDS protocol. In order to better understand and convey the relationships among the many Recommendations that CCSDS has developed, as well as other standards that might be used, effort is underway to establish a methodology for representing end-to-end data system architectures. This methodology, a high level functional architecture, serves several purposes:

• it places each standard in its context to the total data system
• it allows potential users of the standards to quickly identify, for a given domain of interest, those standards that obtain at an appropriate system interface
• it indicates where gaps/overlaps in the total data system development effort may be forming
• it highlights standard interfaces so that modularized functions can be considered
• it promotes industry's interest in developing standard products for these (or other modules) through which the acceptance of standards can be furthered and the benefits of standards can be realized [Ccs].

The approach is to represent the overall system as a small set of boxes in which various high level functions are performed. The boxes are connected by arrows which indicate needed data/metadata flows at the interfaces between the functions. The data/metadata flows can be characterized by the set of standards to which they do/can/must conform. It is worth emphasizing that the model is meant to highlight the methodology first, and only secondarily, the actual functional breakout shown. The model does not represent a rigid architecture that ties the hands of standards developers. Rather, it presents a framework that can be molded to meet developer's views and match manufacturers capabilities. What is perceived for this model is a growing framework, growing in the sense of additional breakouts of detailed functions and services and protocols, that captures the current and future requirements of civil, military, and commercial space endeavor.

Functional model diagram

The diagram below depicts the functional units (rectangles) and interfaces (circles) of the CCSDS Space Data System. This model is a top level model meant to depict the interfaces described in the CCSDS Blue Books. It should be noted that there are other model views used by the CCSDS panels to depict various aspects of the overall space data systems in use by the CCSDS member agencies.

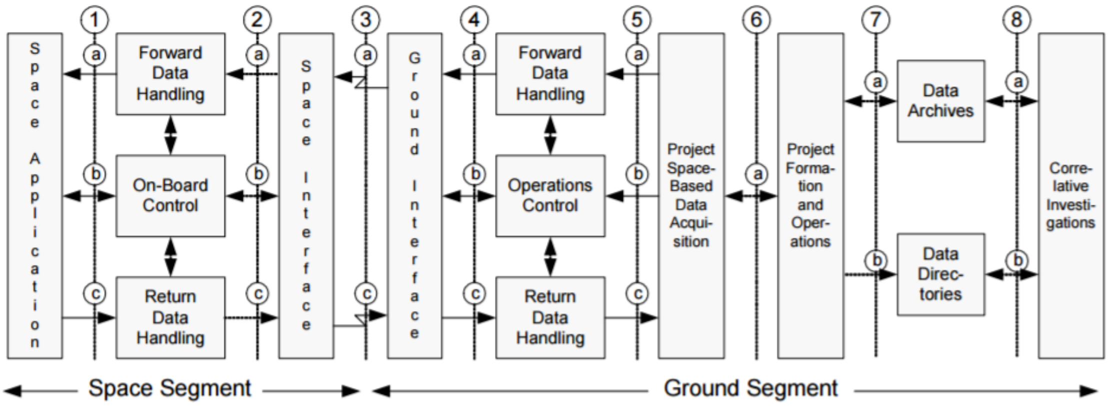

Figure 2.23: Space Data System Functional Model

Protocol Layer Model

The model is a five-layer model sometimes used to represent the protocol layers of Internet-type data communication. It is a simplification of the OSI seven-layer model, with the OSI session and presentation layers omitted. Existing CCSDS Recommendations for space data system protocols and services designed for use on the space link (the communications link between a spacecraft and ground station) are shown in blue. The Internet protocols, in grey, are included to show how they fit into the CCSDS protocol stack when used in conjunction with the CCSDS Link and Physical layers [RVT01]. In addition to the protocols and services identified in the model, the CCSDS is developing a suite of SLE services designed to extend the space link, specifically, to enable spacecraft control and data acquisition from terrestrial locations remote from the ground station terminus of the space link. The organization has also developed Recommendations for a wide array of data description and archiving standards to support end-user handling and storage of telemetry data.

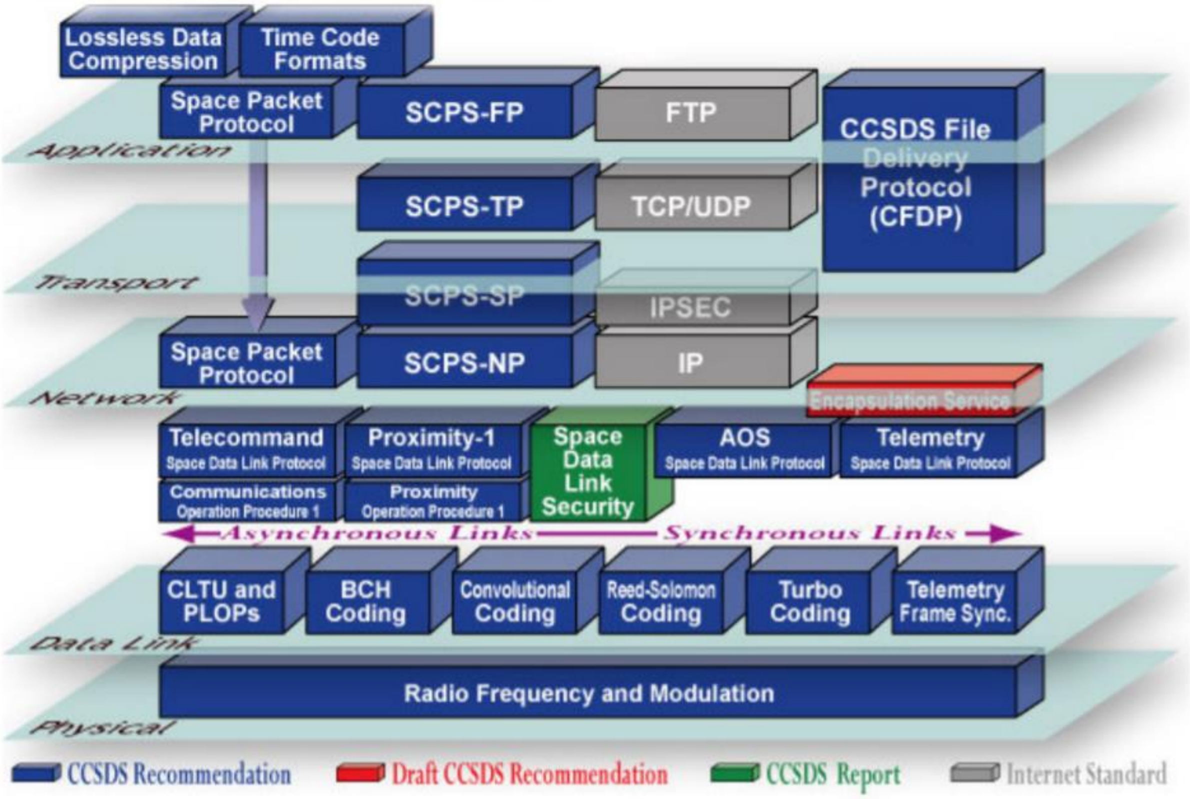

Figure 2.24: Protocol Layer Model

Sources

CCSDS TELEMETRY/TELECOMMAND STANDARDS RESTRUCTURED AS COMMUNICATIONS PROTOCOLS

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.97.2042&rep=rep1&type=pdf