1. Project Context

1.1. STRATHcube

STRATHcube is a student led satellite project at the University of Strathclyde that will be launched from the International Space Station (ISS) with the aim of demonstrating the use of a Passive Bistatic Radar (PBR) for in-orbit detection of space debris. This project aims to create an engineering model of the downlink communication system for the satellite using Commercial Off-The-Shelf (COTS) development boards in compliance with relevant standards.

STRATHcube is being launched as part of the European Space Agency (ESA) Fly Your Satellite (FYS) program. This imposes several requirements on the project such as transmit power limits, control and passivation. The team is also currently working on their application to the FYS Design Booster program, which will provide additional support for the project.

STRATHcube will include a TOTEM Software Defined Radio (SDR) that will handle both the communication and primary payload systems. The TOTEM consists of a Zynq 7020 System on Chip (SoC) connected to an Analog Devices AD9364 RF transceiver with full duplex capabilities. Due to the extremely high cost of space qualified hardware, development boards will be used for the project. A Digilent Zedboard will be used for processing as it has the same Zynq 7020 SoC as the TOTEM, albeit with less memory. The RF transceiver will be modelled using an FMCOMMS1 development board connected via the FMC connector on the Zedboard. This has an AD9361 RF transceiver, which is identical to the AD9364 but with an extra channel that will be disabled.

A research project this summer investigated the configuration of the downlink communication system for the satellite, creating a link budget and selecting Digital Video Broadcasting - Satellite - Second Generation (DVB-S2) as the modulation scheme. The scheme has multiple key advantages, as it has robust Forward Error Correction (FEC) capabilities that allow it to operate extremely close to the Shannon limit for a given Carrier to Noise Ratio (CNR) [1]. Additionally, the Adaptive Coding and Modulation (ACM) capabilities of the system allow it to adapt to changing channel conditions, improving spectral efficiency.

1.2. DVB-S2 Modulation

1.2.1. Overview

The DVB-S2 standard[2] defines the operation of the system, with several optional subsystems depending on the application. The standard defines two types of generic data stream, packetised and continuous. Packetised streams have a fixed size, User Packet Length (UPL), and have Cyclic Redundancy Check 8 (CRC-8) error detection inserted into the beginning of each. Continuous streams include those with variable UPL and do not have CRC-8 inserted.

The data stream is sliced into Data Field Length (DFL) sized chunks and a Baseband Frame (BBFRAME) header is inserted. The BBFRAME is then padded for coding with Bose-Chaudhuri-Hocquenghem (BCH) and Low-Density Parity-Check (LDPC), becoming a Forward Error Correction Frame (FECFRAME). This is then mapped to a constellation, pilots inserted and scrambled to become a Physical Layer Frame (PLFRAME), before finally being filtered and modulated to the carrier frequency.

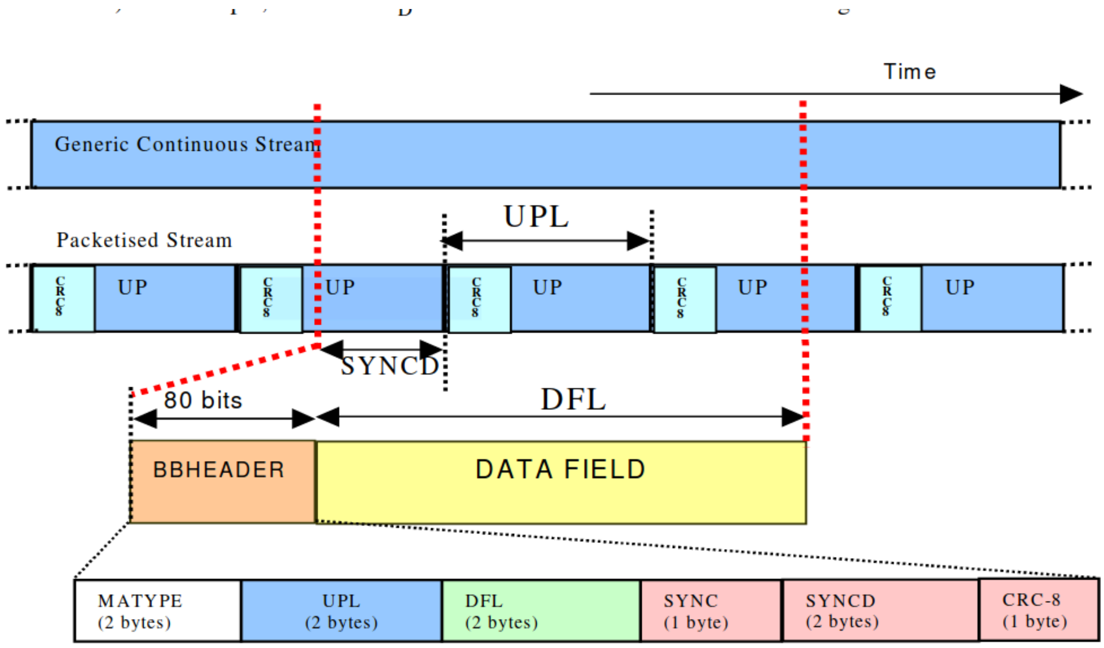


Figure 1: DVB-S2 Slicing Diagram[2]

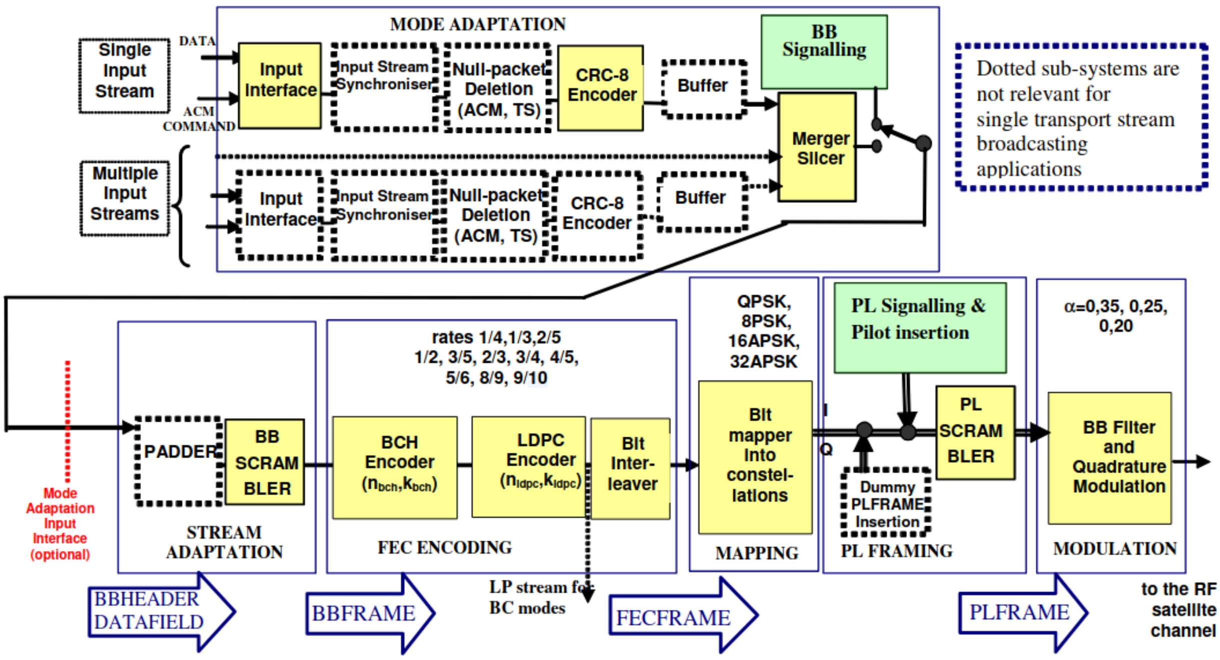


Figure 2: DVB-S2 System Diagram[2]

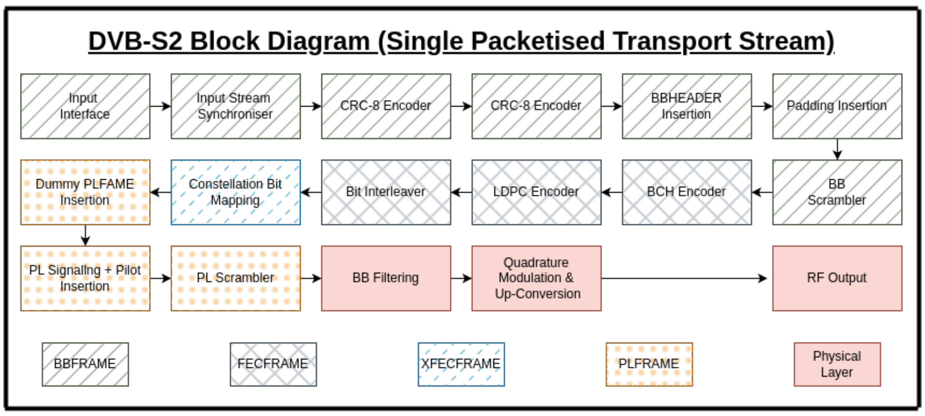


Figure 3: DVB-S2 System Diagram

DVB-S2 uses four modulation schemes, QPSK, 8PSK, 16PSK and 32PSK, coding rates from 1/4 to 9/10 and two FECFRAME lengths, 64800 and 16200 bits, long and short respectively. All of these parameters can be changed from one BBFRAME to the next, allowing for quick adaptation to channel conditions. However, this also introduces an issue with the alignment of packets to frames as a different DFL will be required to minimise padding. This is tackled using Generic Stream Encapsulation (GSE)[3], a link layer protocol that handles the fragmentation of packets based on the current optimal FECFRAME size and DFL. However, this includes several features designed for Internet Protocol (IP) that may not be necessary for the system.

1.2.2. Configuration

As one of the key challenges of this project is managing resource usage, the system configuration will be chosen to minimise the requirements of the PL system. This means that packet scheduling, ACM, GSE and header configuration will be managed by software on the Processing System (PS), alllowing quick reconfiguration of the system in order to optimise for expected channel conditions.

GSE implementation will be investigated, but a simpler system may be used to reduce complexity. This will be decided based on the availability of open-source implementations and time availability.

1.2.3. ACM Analysis

Due to the complexity of implementing ACM in the system, an investigation into the benefits over a fixed configuration that maximised availability was conducted. The satellite was simulated for the full mission duration at its initial altitude of 400km using the MATLAB Satellite Communications Toolbox. The elevation of the satellite during each pass over the ground station was binned into 5° increments. An existing link budget was used to calculate the CNR for each elevation bin and cross-referenced to the optimimum modulation and coding rate using the DVB-S2 standard. The maximum possible throughput was calculated for each elevation bin and multiplied by the time spent at each elevation to find the total throughput for each strategy. The results are shown in Figure 4 and show a significant increase in throughput, proving that the complexity of implementing ACM is worth the benefits even despite the simplicity of the simulation.

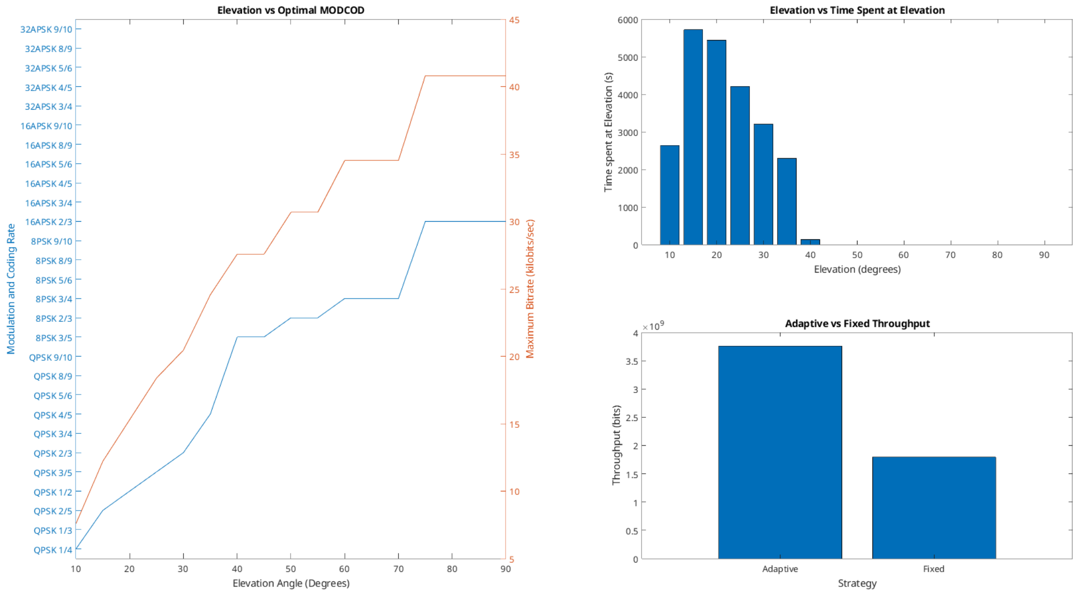


Figure 4: Left: Elevation vs Modulation and Coding Rate (Blue), Modulation and Coding rate to throughput (red). Top Right: Time spent in each elevation bin. Bottom Right: Throughput vs strategy.

)

1.3. Packet Handling

1.3.1. XTCE

The XML Telemetric and Command Exchange (XTCE) standard[4] allows consistent definition of data packets, algorithms and commands to be used within space systems. This has been created by the Consultative Committee for Space Data Systems (CCSDS) in order to promote inter-operability and has has been selected for the project. This allows definition of generic packets for development without restricting the final system.

As XTCE is quite a complex standard so a NASA tool called Core Flight System (CFS) Command and Data Dictionary (CCDD) will be used to manage packet definitions. This can also be used to generate C headers for parsing, or an existing Python library may be used. The Python approach could be simpler to implement, but the C headers will be more efficient and allow for easier integration with the rest of the system.

1.3.2. Interfacing

The TOTEM SDR has multiple interfaces to other systems. The main ones for flight are Controller Area Network (CAN) and Inter-Integrated Circuit (I2C), with Ethernet included for ground testing. The system may also have to send packets from other programs running on the PS. This will be achieved using a First In First Out (FIFO) buffer with Linux Inter-Process Communication (IPC) mechanisms.

1.3.3. Scheduling & Compression

The onboard data handling has been analysed in a previous dissertation project. This identified the need to prioritise housekeeping data, such as GNSS data and power telemetry, over primary payload data. This project also defined multiple packet structures, along with their priority. These will be used to create sample packets for testing of the system.

This project also identified the need for compression of the data to reduce the amount to be transmitted, with the lossless POCKET algorithm being selected for this purpose. A C++ implementation of the newer POCKET+ algorithm is available that has been created based on the CCSDS standard and will be used on the PS.

1.3.4. ACM Router

The ACM router is responsible for managing the configuration of the DVB-S2 system. This includes selection of modulation, coding rates, FECFRAME length and DFL. In the final system these decisions will be made based on information received from the ground station, however for this project they will be made based on a pre-defined schedule. This will allow for the system to be tested under a variety of conditions and for the system to be optimised for expected channel conditions.

1.3.5. PL Interfacing

This block will handle the alignment of packets to data fields and pass this to the Programmable Logic (PL), along with control data. The interface driver will be created using the libiio library created by Analog Devices.

1.4. System Architecture

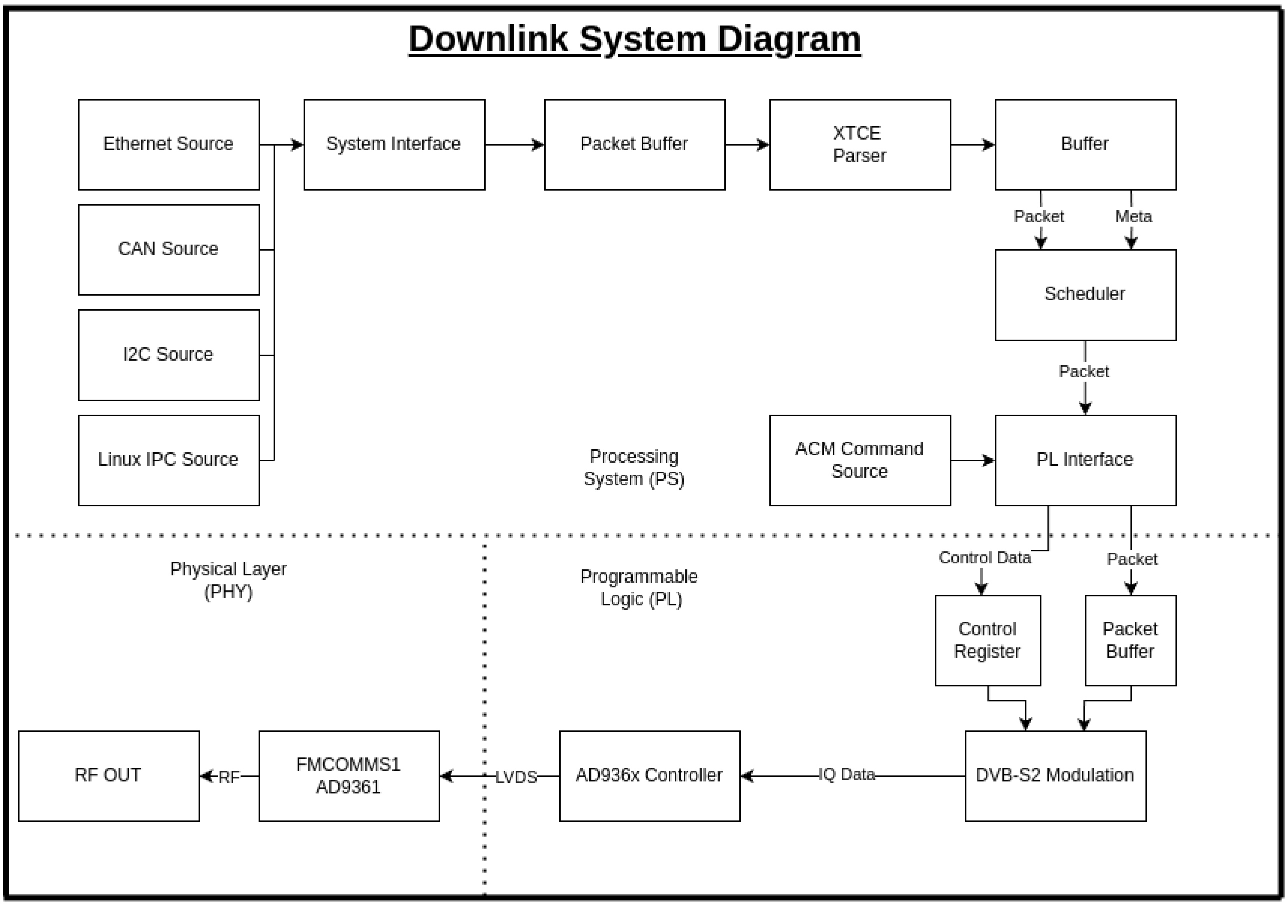
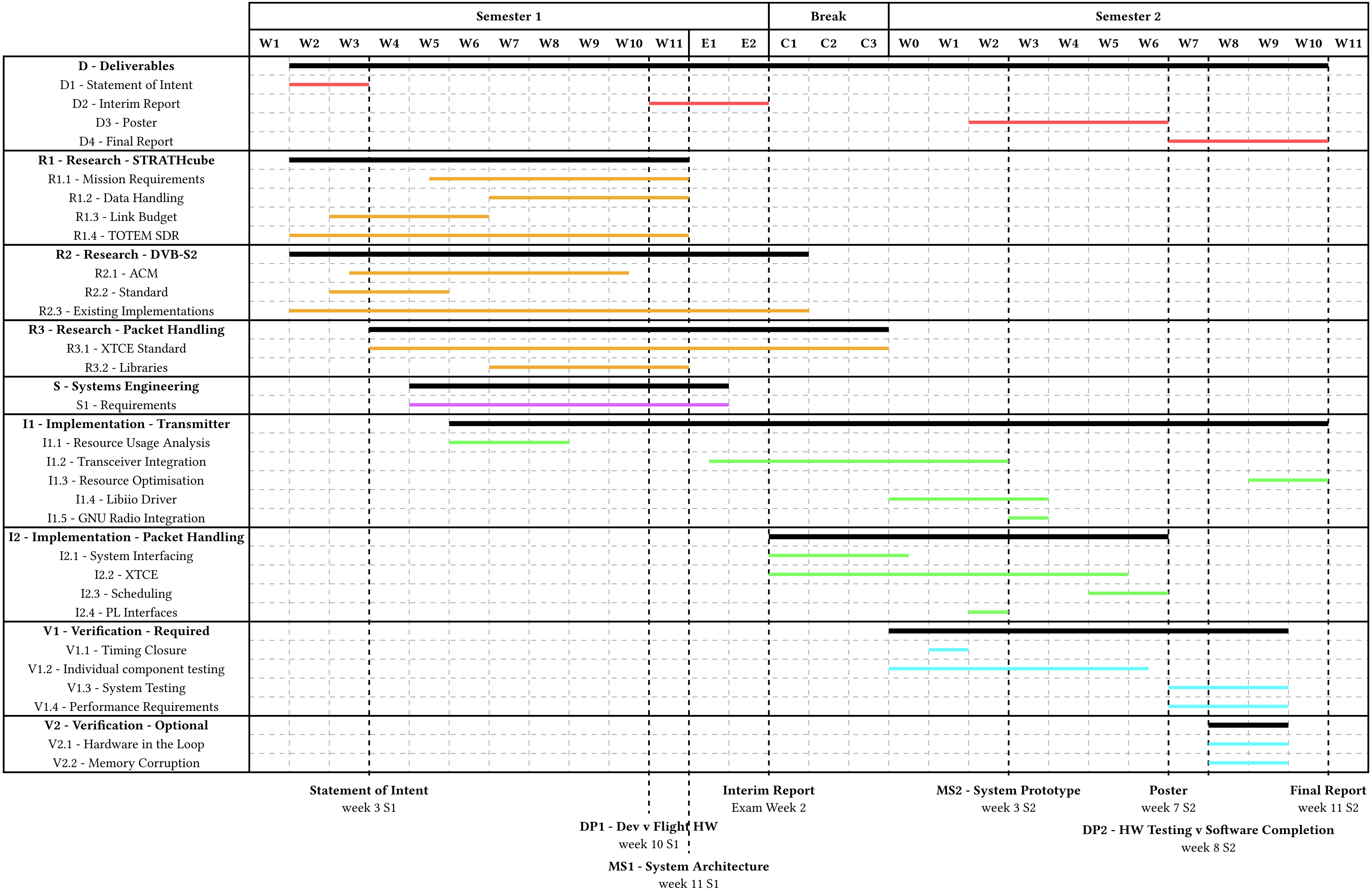


Figure 5: System Diagram

2. Project Timeline



3. Project Review

3.1. Project Plan

The first semester has primarily been devoted to the research and planning of the project as this was an area that had not been fully investigated prior. The main outcome of this work has been an in-depth system architecture that will facilitate implementation efforts in semester two. The lack of December exams also provides a key opportunity to get started on development before returning to classes in January.

There are three key milestones outlined in the project timeline. The first being the system architecture, which has been defined in Section 1. The second is the creation of a "System Prototype" which will demonstrate the minimum features necessary to send a packet through the system. This is targeted for Week 3 of Semester 2. The final milestone is the report submission. This will mark the completion of the system and documentation of the project.

There are two key decision points, marked "DP1" and "DP2". DP1 has been reached, and marks the decision to target development hardware, as the TOTEM SDR could not be procured in time for the project. DP2 will be reached in Week 8 of Semester 2 and will mark the decision to focus on end to end testing of the system, rather than feature development and completeion of software and RTL components.

To ensure time is left for the final report, a design freeze will be implemented during week 9 of semester 2. This means that no major features or changes will be implemented from that point onwards to ensure that the system is finalised and tested in advance of the final report deadline.

3.2. Technical Risks

Risks	Mitigation	Likelihood	Severity	L*S
Catastrophic Dataloss	1. Use of cloud storage 2. Use of Git source control 3. Regular snapshots of work	1	5	5
Severe Illness	1. Creation of draft submissions on Myplace to avoid missing deadlines 2. Report personal circumstances on Pegasus 3. Discussion of reduction of scope with supervisor	1	4	4
Breakage of key equipment	1. Only work in designated areas 2. Use of equipment as designed, with care given for fragile connectors 3. Identification of alternatives available in the department	2	3	6
"Feature Creep"	1. Creation of requirements document 2. Regular review of work with supervisor 3. Scope freeze decided at Interim report, with options for further reduction identified in project plan	3	2	6
Difficulty in system integration	1. Regular project meetings with supervisor 2. Adherence to best practices in RTL design with continous verification and testing 3. Use of version control to track changes 4. Use of a modular design to allow verification of components 5. Use of resources from manufacturers to ensure compatibility	3	2	6
Extended learning curves of new languages and tools	1. Use of online resources 2. Use of existing knowledge of similar tools to reduce learning curve 3. Use of manufacturer resources	3	2	6
System not fit for purpose	1. Design with requirements in mind 2. Use of open-source examples to ensure compatibility and reliability 3. Use of a modular design to allow for easy changes	2	3	6

3.3. Sustainability

STRATHcube's primary mission to demonstrate the use of a PBR is also combined with the secondary mission to collect telemetry data during re-entry to tackle some of the most important issues in the space industry today. The increasing number of satellites in orbit is leading to a large amount of space debris which poses a significant risk to operational satellites in congested areas such as sun synchronous orbits. This field is referred to as space situational awareness. The ability to detect and track space debris is crucial to ensure the safety of future missions however there are resolution issues with ground based solutions that can be solved by using a PBR in orbit.

The main solution today for space debris is to de-orbit satellites at the end of their operational life. This process is difficult to model at present, as there are complex interactions between the satellite and the atmosphere at hypersonic speeds. The data collected during the re-entry of STRATHcube will be used to validate TITAN, a tool developed by the University of Strathclyde and overall improve the process of "Design for Demise" which is an increasingly important requirement for satellite design.

3.4. Ethical Considerations and EDI

The main ethical considerations for this project lie in compliance to relevant telecommunications regulations from the International Telecommunication Union (ITU) and Office of Communications (OFCOM). During development, all transmissions will be conducted using certified equipment and through cables to ensure that no harmful interference is caused.

The resulting source code will be made available to the public under an open-source license to ensure that the project is transparent and accessible to all. This will also allow for the project to be continued for further use in STRATHcube or as a resource for other projects.

Aditionally, space is a resource that underpins modern society and it is important to ensure that it is used responsibly. The primary and secondary missions of STRATHcube will positively contribute to space sustainability and as such will have a positive impact that will support society as a whole.

4. References

[1] M. Eroz, F.-W. Sun, and L.-N. Lee, "DVB-S2 low density parity check codes with near Shannon limit performance," *International Journal of Satellite Communications and Networking*, vol. 22, no. 3, pp. 269–279, 2004, doi: 10.1002/sat.787.

[2] "DVB-S2 Standard."

[3] "TS 102 606-1 - V1.2.1 - Digital Video Broadcasting (DVB); Generic Stream Encapsulation (GSE); Part 1: Protocol."

[4] "XML Telemetric and Command Exchange—Version 1.2," 2020.

[5] "MATLAB DVB-S2 HDL Transmitter." Accessed: Dec. 03, 2024. [Online]. Available: https://uk.mathworks.com/help/satcom/ug/dvbs2-hdl-transmitter.html

[6] "gr-dvbs2rx GitHub Repository." Accessed: Dec. 03, 2024. [Online]. Available: https://igorauad.github.io/gr-dvbs2rx/

[7] Gavin Medley, Michael Chambliss, and Greg Lucas, "space_packet_parser." Accessed: Dec. 03, 2024. [Online]. Available: https://github.com/medley56/space_packet_parser

[8] K. Deergha Rao, "Linear Block Codes," *Channel Coding Techniques for Wireless Communications*. Springer India, New Delhi, pp. 73–126, 2015. doi: 10.1007/978-81-322-2292-7_4.

[9] A. Morello and V. Mignone, "DVB-S2: The Second Generation Standard for Satellite Broad-Band Services," *Proceedings of the IEEE*, vol. 94, no. 1, pp. 210–227, Jan. 2006, doi: 10.1109/JPROC.2005.861013.

[10] "nasa/CCDD." Accessed: Dec. 10, 2024. [Online]. Available: https://github.com/nasa/CCDD

[11] D. J. Evans and A. Donati, "The ESA POCKET+ housekeeping telemetry compression algorithm: Why make spacecraft operations harder than it already is?," in *2018 SpaceOps Conference*, Marseille, France: American Institute of Aeronautics, Astronautics, May 2018. doi: 10.2514/6.2018-2613.