

Lunar CubeSat Software Architecture Analysis

Rukmal Weerawarana

Department of Finance and Business Economics

University of Washington

June 3, 2015

Author Note

Summary of research conducted during the ESS 490 course (Spring Quarter 2015) as a part of the Advanced Propulsion Laboratory's CubeSat Program in the Earth & Space Sciences Department of the University of Washington

Table of Contents

Table of Contents	2
Abstract	3
Core System Requirements and Considerations	4
Resource Limitations and Possible Solutions	4
Process scaling.....	4
Communication system optimizations.	4
Data preprocessing and compression.	5
Communications System	6
Selective Data Compression.....	6
Data compression standards.....	7
ICER image compression.	7
Client – Server Data Request System	7
Transmission Redundancy Measures	8
Simple repetition coding.....	9
Linear block coding.....	9
Conclusion and Further Investigation	10
References.....	11

Abstract

As this is one of the first missions of its kind, a Lunar CubeSat faces challenges that have not been addressed at this scale before. Following this premise, I attempted to focus my research on the Software Architecture challenges faced by the system, honing in on the major issues raised when designing the software running on the Flight Computer and the Communications system architecture. I proposed a variety of possible software architecture paradigms that could be incorporated into the final flight computer system, including process scaling techniques, and the possibility of leveraging data compression to reduce storage use on the system. In terms of the communications challenges, I proposed a client-server architecture that could save significant power for the system. I also detail possible redundancy measures that could be adopted to reduce the chance of data being corrupted due to transmission issues.

Core System Requirements and Considerations

After careful consideration of past CubeSat missions and with particular consideration to the research done by the Avionics team of the previous quarter¹ (Winter 2015), the following core system requirements were established for the Flight Computer.

- Run a high-availability operating system that has functionality to allocate resources independently to individual processes.
- Have relatively low power consumption, due to the limited power resources available to the on-board computer on the CubeSat.
- Support common data-transference protocols for data transfer between subsystems on the CubeSat. Possible standards for use with data transference include TCP & UDP IP protocols² and Unix/IPC sockets.³
- Continually capture data from on-board sensors and store raw data to a data storage medium, and possibly change data gathering configuration as per configuration updates from ground control.
- Have the ability to be rebooted on the command of a watchdog timer, or some other such device that will help combat radiation-related software crashes. Ideally, the system would also have the ability to transmit error logs to ground control on command for further analysis.

Resource Limitations and Possible Solutions

Due to the limited resources available to the on-board computer on the CubeSat in terms of power, certain trade-offs may have to be made in order to keep power consumption within optimal levels. Some of the main trade-offs that will be made in terms of software are discussed below.

Process scaling. Computer systems often scale back processes to reduce power consumption overall. As it would not be advisable to let the operating system automatically decide which processes to scale back (for example, it would not be ideal to scale back attitude control systems, but it would be acceptable to reduce image capture rates with the camera while power levels remain low).

Thus, a process-ranking system would be ideal for this purpose, as it would be possible to configure a custom process-hierarchy that can be used by the power-management system.

Communication system optimizations. Another possible area that could be optimized to reduce overall power consumption would be the communications system. Typically, communications systems on deep-space craft are major sources of power consumption, due to the high-accuracy and data-intensive nature of the transmissions.⁴

Power consumption and resource use by communications systems could be reduced by utilizing a batch-transmission strategy where a large amount of compressed data is transmitted to ground control

¹ (Harrigan, 2015)

² (Computer Engineering Students of San Jose State University, 2015)

³ (man-pages project, 2015)

⁴ (Manshadi, 2012)

at once, as opposed to continually streaming data from the satellite. This would reduce both stress on the computer and the power consumption of the communications subsystem present on the CubeSat.

Data preprocessing and compression. The CubeSat will be collecting large amounts of data from a wide variety of sensors, ranging from thermometers to positional data from the craft's various sensors. Data preprocessing could potentially be performed on this data using the on-board computer to determine which data will eventually be stored and/or sent back to ground control using the communications subsystem. Algorithms for such preprocessing are already in existence and were used on the Chinese *Chang'e3* Lunar Lander.⁵

Furthermore, data compression techniques will be a strategy that – if utilized effectively – will be extremely valuable for conserving data storage space on board the craft. As the CubeSat will be capturing image data, combinations of lossless and lossy compression (lossless compression does not compromise data, whereas lossy compression reduces the quality of the data) techniques⁶ can be used in unison to reduce file size significantly.

⁵ (Wen, et al., 2014)

⁶ (Higher National Computing, 2008)

Communications System

Due to the autonomous nature of the mission, the integrity and performance of the communications system will be of utmost importance. It will be used both for information retrieval from the craft, and also to give the craft instructions with regards to communication rendezvous points, and possibly even deliver software updates to the Flight Computer.

Communications in deep space is extremely challenging. The performance of deep space communication systems typically degrade with inverse proportionality to the square of the distance to Earth.⁷ Thus, if the difficulty of communicating with a geostationary satellite had a baseline difficulty of 1, communication with a satellite in orbit around the Moon would be approximately 100.

This staggering increase in difficulty compared to typical CubeSat missions calls for major changes to how the problem of communication with the satellite is approached. Not only is the performance of the downlink decreased significantly, but scattering of the signals over the distance between the Earth and Moon would also cause a significant decrease in the data rate of the communications link.

Possible improvements to address each of the issues detailed above are discussed further below.

Selective Data Compression

Assuming that by the time the CubeSat is launched it is equipped with a 4K (Ultra High Definition) camera, a thermometer and an accelerometer, the computer would be capturing a vast amount of data daily. Additionally, ground control would also be interested in downloading log files of the flight computer for diagnostic purposes. Due to the high data to downlink ratio, it would be extremely impractical if not impossible to beam all of the data down from the satellite in its original form.

Understandably, some of the data – mainly logs from the flight computer and positional data – are extremely valuable in their original form. Other, more cosmetic data such as images from the camera would not need to be downloaded at their original quality. This differentiation between the importance of different data can be used to alternate between lossy and lossless techniques when compressing the data prior to beaming it back to ground control.

Furthermore, due to the very high risk of losing packets of data during transmission back to Earth, it is vital that a progressive compression technique is utilized,⁸ as opposed to a non-progressive compression algorithm. In layman's terms, progressive compression algorithms first store an 'overview' of the data in compressed form, and then add detail as the compression progresses. This allows for the possibility of reconstructing the data at ground control even if some of it is lost during transmission.⁹

⁷ (Manshadi, 2012)

⁸ (Zhang, Jing, & Shaochuan, 2010)

⁹ (Waschbüsch, Gross, Eberhard, Lamboray, & Würmlin, 2004)

Data compression standards. The Consultative Committee for Space Data Systems (CCSDS¹⁰) is the multi-national forum for the development of communications and data systems standards for spaceflight. Currently, they recommend the use of a simple lossless compression algorithm,¹¹ similar to Huffman Coding¹² for the compression of simple data, such as diagnostic information from the flight computer, and raw data from the sensors on board the CubeSat. This compression strategy can also be used to compress images a second time after they have been captured by the camera.

ICER image compression. The ICER image compression standard was developed at NASA's Jet Propulsion Laboratory (JPL) as an alternative progressive compression algorithm that is less processor-intensive than the popular JPEG2000 image format.¹³ This format was specifically designed for the purpose of deep space transmission. It is extremely efficient, and is currently used on the *Curiosity* and *Opportunity* Mars rovers.¹⁴ This format will also be particularly useful in the event that the CubeSat eventually includes a multi-spectral camera, such as an Infrared (IR) camera, as it supports Hyperspectral (multi-spectral) image capture.¹⁵ Furthermore, it also has lossy and lossless compression modes, which gives ground control more versatility when requesting image transmission from the CubeSat.

Client – Server Data Request System

Due to the limited power resources available to us on the CubeSat, it is of utmost importance that we conserve as much power as possible. Researchers in the field of fault-tolerant computing have reached general consensus that this system is the most reliable compromise between reliability and power conservation.¹⁶

The usage of this system would yield many benefits for the Lunar CubeSat mission. Due to the fact that the CubeSat will be orbiting the moon, the probability of accurately scheduling rendezvous points between the ground control, and guaranteeing that they would occur at times of good weather is extremely slim. Using this architecture will allow us to control the times at which the CubeSat relays data back to ground control, without the CubeSat constantly using its resources in an attempt to relay data. This is doubly important considering that ground control will be based in Seattle, Washington where the weather is often unfavorable for strong satellite connections.

¹⁰ (Consultative Committee for Space Data Systems, 2015)

¹¹ (CCSDS Secretariat, 2013)

¹² (Lee, 2011)

¹³ (Kiley & Kilmesh, 2003)

¹⁴ (Kiley & Kilmesh, Preliminary Image Compression Results from the Mars Exploration Rovers, 2004)

¹⁵ (Kiley, Kilmesh, Xie, & Aranki, 2006)

¹⁶ (Huang & Kintala, 1993)

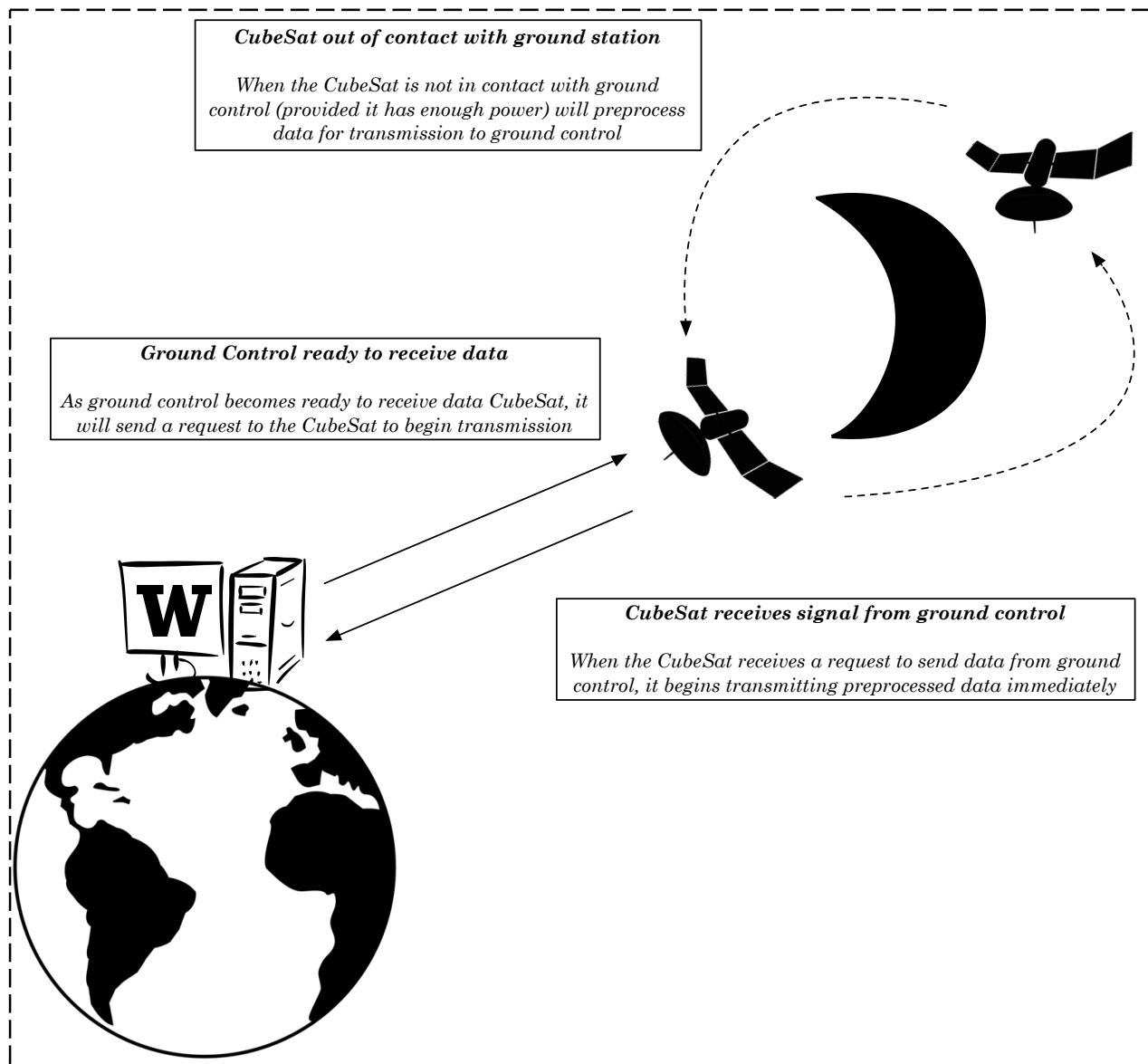


Figure A Client - Server sample request workflow

Furthermore, the implementation of this system would free up the flight computer to preprocess data (error-check, compress etc.) that can be sent as soon as it receives a signal from ground control. This process is illustrated in Figure A above. This could potentially reduce wait time for ground control, as data will be ready to be transferred down as soon as the CubeSat receives the signal.

Transmission Redundancy Measures

As radio waves from the CubeSat travelling to Earth must traverse an extremely large distance to reach its destination, the radio signals will undergo significant free space path loss due to the natural diffraction of the wave as it travels through space.¹⁷ This phenomenon significantly reduces the intensity of the signal received at the ground station relative to that originally emitted by the flight

¹⁷ (The Institute of Electrical and Electronics Engineers, Inc., 2004)

computer.

Due to the reduced intensity, the probability of losing data during transmission increases significantly. There are many strategies that can be implemented to reduce the probability of total data loss, despite this unavoidable drop in signal intensity. A combination of the methods discussed below could be utilized to increase the likelihood of successful data transmission.

Simple repetition coding. Repetition coding is often described as one of the most basic error-correcting algorithms. It is based on the concept of re-transmitting the same segment of a message a pre-determined number of times. The receiver could then recover the original message by looking at the received messages, and determining which sequence was received at the highest frequency.

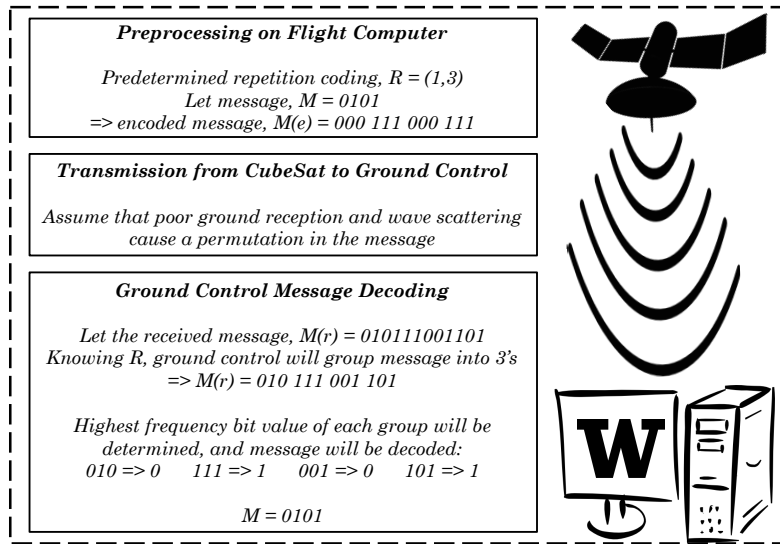


Figure B Simple repetition coding process workflow

As the probability of a permutation occurring at the same point in the message decreases proportionally to the number of times the message is repeated, this method is particularly effective in eliminating random errors that would ordinarily be caused by the scattering of the transmission wave.¹⁸ The trade-off with this particular strategy would be the performance impact, as it is highly inefficient to repeat the same message multiple times over – especially when the volume of data being transmitted is large.

Linear block coding. Hamming codes are linear error-correcting codes that can be used to detect up to two-bit errors and correct single bit errors without detection of uncorrected errors.¹⁹ Hamming codes use concepts of bit padding, which involves adding extra bits to a small chunk of data to increase data integrity. These *parity bits* can then be used to help determine the validity of the remaining data bits when decoding the received message. Additionally, the use of Hamming codes will also help combat the issue of potential bit-flips (0 to 1 and vice versa) caused by Gamma radiation.

¹⁸ (Fan, Wang, Thompson, & Poor, 2007)

¹⁹ (Hamming, 1950)

Conclusion and Further Investigation

When beginning this research, I sought to address some of the main issues posed by putting a CubeSat in orbit around the moon. I focused my investigation on the issues surrounding the software that would be run on the flight computer, and tackled challenges posed by the increased difficulty of communication with the CubeSat.

Considering the requirements for the system – which included high-availability and support for common data-transference protocols – I proposed some optimizations that could be made to the Flight Computer. The first of the improvements was utilizing a technique called *process scaling*, which could potentially be used to better allocate system resources based on priority and power consumptions levels on the CubeSat. To address the problem of data storage limits, I proposed alternating between lossless and lossy compression algorithms to reduce the file sizes of captured data.

In terms of the communications system, I proposed utilizing a client-server system which would shift the point of initialization of a communications stream from the CubeSat to ground control. This could potentially yield large savings in terms of power and data loss, as it would ensure that ground control is indeed receiving data, and the CubeSat is not aimlessly transmitting data into space. To reduce the amount of data transmission required, I also explored specific compression techniques that could be used for images. Furthermore, I also proposed transmission redundancy measures that could be implemented to reduce the probability of errors caused during data transmission interfering with data collection. These measures included techniques used by most satellite systems today, specifically simple repetition coding and linear block coding techniques.

In terms of further study, I recommend that the next step in addressing the issues detailed in this report would be to test some of the concepts proposed. Additionally, the fragile state of the Flight Computer (i.e. expected restarts) should also be taken into account when designing other mission-critical systems such as propulsion and navigation.

In terms of testing the concepts detailed in this report, one possible approach could be to use a programming model in conjunction with power consumption levels of the communications system and estimations of bit-flips and other such signal modulations to predict power levels required to transmit data back to Earth. This information could then be used to improve on the System Architecture recommendations made in the report before finally moving on to implementing the system for the Lunar CubeSat launch.

References

- CCSDS Secretariat. (2013). *Lossless Data Compression Informational Report*. The Consultative Committee for Space Data Systems. Washington, DC: CCSDS Management Council.
- Computer Engineering Students of San Jose State University. (2015, 5 28). *Communication Networks/TCP and UDP Protocols*. Retrieved 6 1, 2015, from Communication Networks - Wikibooks: http://en.wikibooks.org/wiki/Communication_Networks/TCP_and_UDP_Protocols
- Consultative Committee for Space Data Systems. (2015, 6 3). *Consultative Committee for Space Data Systems (CCSDS) Homepage*. Retrieved 6 3, 2015, from <http://public.ccsds.org/default.aspx>
- Fan, Y., Wang, C., Thompson, J., & Poor, H. V. (2007). Recovering Multiplexing Loss through Successive Relaying Using Repetition Coding. *IEEE Transactions on Wireless Communications* , 6 (12), 4484-4493.
- Hamming, R. W. (1950). Error Detecting and Error Correcting Codes. *The Bell System Technical Journal* , 29 (2), 147-160.
- Harrigan, C. (2015). *Lunar CubeSat design using non-radiation-hardened components*. University of Washington, Departments of Mathematics & Computer Science. Seattle: University of Washington.
- Higher National Computing. (2008, 8 6). *Compression Techniques*. (Scottish Qualifications Authority) Retrieved 6 1, 2015, from Higher National Computing: E-Learning Materials: http://www.sqa.org.uk/e-learning/BitVect01CD/page_82.htm
- Huang, Y., & Kintala, C. (1993, 6). Software Implemented Fault Tolerance: Technologies and Experience. *Proceedings of 23rd Intl. Symposium on Fault Tolerant Computing (FTCS-23)* , 2-9.
- Kiley, A., & Kilmesh, M. (2004). *Preliminary Image Compression Results from the Mars Exploration Rovers*. National Aeronautics and Space Administration (NASA), Jet Propulsion Laboratory (JPL). Pasadena, CA: NASA JPL Interplanetary Network Directorate (IND).
- Kiley, A., & Kilmesh, M. (2003). *The ICER Progressive Wavelet Image Compressor*. National Aeronautics and Space Administration (NASA), Jet Propulsion Laboratory (JPL). Pasadena, CA: NASA JPL Interplanetary Network Directorate (IND).
- Kiley, A., Kilmesh, M., Xie, H., & Aranki, N. (2006). *ICER-3D: A Progressive Wavelet-Based Compressor for Hyperspectral Images*. National Aeronautics and Space Administration (NASA), Jet Propulsion Laboratory (JPL). Pasadena, CA: NASA JPL Interplanetary Network Directorate (IND).
- Lee, J. (2011). Huffman Data Compression. *MIT Undergraduate Journal of Mathematics* .
- man-pages project. (2015, 5 7). *Linux man-pages: unix - sockets for local interprocess communication* . Retrieved 6 1, 2015, from Linux Programmer's Manual: <http://man7.org/linux/man-pages/man7/unix.7.html>
- Manshadi, F. (2012). *Deep Space Communication*. National Aeronautics and Space Administration

- (NASA), Jet Propulsion Laboratory (JPL). Pasadena, CA: NASA Technical Reports Server (NTRS).
- The Institute of Electrical and Electronics Engineers, Inc. (2004). *IEEE Standard Definitions of Terms for Antennas*. New York, NY: Antenna Standards Committee of the IEEE Antennas and Propagation Society.
- Waschbüsch, M., Gross, M., Eberhard, F., Lamboray, E., & Würmlin, S. (2004). Progressive Compression of Point-Sampled Models. *Eurographics Symposium on Point-Based Graphics* , 95-102.
- Wen, W.-B., Wang, F., Li, C.-L., Wang, J., Cao, L., Liu, J.-J., et al. (2014). Data preprocessing and preliminary results of the Moon-based Ultraviolet Telescope on the CE-3 lander. *Research in Astronomy and Astrophysics* , 14 (12), 1674-1681.
- Zhang, J., Jing, M., & Shaochuan, W. (2010). Robust transmission of progressive images in the deep space communication. *2010 IEEE Conference on Wireless Communications, Networking and Information Security (WCNIS)* , 4-8.