CubeSat Networks: Balancing Power with Satellite-to-Ground Data Throughput

by

Stephen Ennis

Supervised by Dr. Jonathan Dukes

Presented to the University of Dublin, Trinity College

In partial fulfilment of the requirements for the degree of

M.A.I. Computer Engineering

Submitted May, 2017

Declaration

I, Stephen Ennis, declare that the following dissertation, except where otherwise stated, is entirely my own work; that it has not previously been submitted as an exercise for a degree, either in the University of Dublin, Trinity College, or in any other University. I agree that the University of Dublin, Trinity College may lend or copy the following dissertation, or any part thereof, upon request.

Stephen Ennis

May 18, 2017

Acknowledgements

I would like to thank Dr. Jonathon Dukes for his guidance throughout my dissertation. Particular thanks are due as Dr. Dukes voluntarily took on the supervision of my dissertation project during a notably busy period of the academic term. Working with Dr. Dukes has proved continually enriching and enjoyable.

I would also like to thank Dr. Stephen Farrell and Dr. Ciaran McGoldrick of Trinity College Dublin for their support and guidance during the early stages of this work. Thanks are also due to my previous mentors, Dr. Paul Brenner and Dr. Sunny Shah of the University of Notre Dame for their assistance in various research endeavours which have led to this work.

This work is inspired by the achievements of the engineers, scientists, operators and researchers behind missions such as NASA's Nodes and EDSN missions, ESA's GomX-3 mission and the Tianwang-1 mission. The opportunity to one day contribute to mankind's understanding of the universe and our place within it, as such individuals have, provides a constant source of motivation.

Finally, I am deeply grateful to my Mother, Áine Ennis, for her constant support and encouragement throughout my studies.

Abstract

CubeSats are small satellite platforms which have significantly reduced the cost of access to low Earth orbit over the past decade. Recent CubeSat missions have demonstrated the platform's ability to form in-orbit networks. CubeSat Network (CSN) missions enable low-cost applications in coordinated sensing and low-bandwidth communications.

This work addresses a trade-off unique to CSNs. CubeSat satellite-to-ground (S2G) communication requires high levels of energy consumption to achieve data rates in the order of kilobytes per second. In comparison, CubeSats are capable of more energy efficient satellite-to-satellite (S2S) communication at rates an order of magnitude above those of S2G communication. This asymmetry underpins this work's trade-off of interest, that of CSN power use against S2G data throughput.

Relevant areas of prior art are examined and specialized Medium Access Control (MAC) and routing protocols are proposed. This work's proposed protocols are developed alongside a simulation of a hypothetical CSN mission using the open-source network simulator, OMNeT++. Proposed MAC protocol energy saving features are shown to decrease CSN energy consumption without a reduction in S2G throughput. This work's proposed routing protocol introduces the energy sensitive election of a CubeSat dedicated to performing S2G communication. This election approach is shown to reduce the energy consumption of previously "over-worked" CubeSats. Additional adjustments to route discovery behaviour are required to ensure this approach does not reduce the overall energy efficiency of S2G communication.

Contents

Figures

Tables

Abbreviations

ACK	Acknowledgement
ADCS	Attitude Determination and Control Systems
ADS	Automatic Dependent Surveillance
AIAA	American Institute of Aeronautics and Astronautics
AIM	Asteroid Impact Mission
AODV	Ad hoc On-demand Distance Vector
ARQ	Automatic Repeat Request
AX.25	Amateur X.25
BGP	Border Gateway Protocol
BLE	Bluetooth Low-Energy
BPSK	Binary Phase Shift Keying
C&DH	Command and Data Handling
C/TDMA	Code/Time Division Multiple Access
C2CNet	Car-to-Car Network
CARUS	Cooperative Autonomous Reconfigurable UAV Swarm
CBRP	Cluster Based Routing Protocol
CDMA	Code Division Multiple Access
CeREs	a Compact Radiation belt Explorer
CLO	Cross Layer Optimization
CNSA	Chinese National Space Agency
COPINS	CubeSat Opportunity Payloads
COTS	Commercial Off The Shelf
CPOD	Cubesat Proximity Operations Demonstration
CSMA	Carrier Sense Multiple Access
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CSN	CubeSat Network
CTS	Clear To Send
CZ-2D	Chang Zheng-2D
D^3	DYMO Cubed
DARPA	Defense Advanced Research Projects Agency

DICE	Dynamic Ionosphere CubeSat Experiment
DSR	Dynamic Source Routing
DTN	Delay Tolerant Network
DYMO	Dynamic MANET On-demand routing protocol
EARM	Energy-Aware Routing to Mobile gateway
EDSN	Edison Demonstration of Smallsat Networks
ELaNa	Educational Launch of Nanosatellites
EO	Earth Observation
ER	Energy Rank
ERI	Earth Radiation Index
ESA	European Space Agency
FANET	Flying Ad-Hoc Network
FDD	Frequency Division Duplex
FDMA	Frequency Division Multiple Access
FIFO	First In First Out
G2S	Ground To Satellite
GM	Ground Master
GNSS	Global Navigation Satellite System
GPS	Global Positioning Satellite
GPSR	Greedy Perimeter Stateless Routing
HCL	High Contention Level
I/F	Interface
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IPv4	Internet Protocol version 4
ISARA	Integrated Solar Array and Reflectarray Antenna
ISS	International Space Station
JPL	Jet Propulsion Laboratory
LAICE	Lower Atmosphere/Ionosphere Coupling Experiment
LCL	Low Contention Level
LDMA	Load Division Multiple Access
LEO	Low Earth Orbit
LMAC	Lightweight Multiple Access
MAC	Medium Access Control

MANET	Mobile Ad-Hoc Network
MASP	Maximum Amount Shortest Path
MIMO	Quadrature Phase Shift Keying
MLSP	Multi-Layered Satellite Routing
MOO	Multi-Objective Optimization
NASA	National Aeronautics and Space Administration
ND	No Data
NIC	Network Interface
NS-3	Network Simulator - 3
OLFAR	Orbiting Low Frequency Antennas for Radio Astronomy
OLSR	Optimized Link State Protocol
OMNeT++	Objective Modular Network Testbed in C++
OSI	Open Systems Interconnection
PFF	Precision Formation Flying
PvTP	Power versus Throughput
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
RAVAN	Radiometer Assessment using Vertically Aligned Nanotubes
RERR	Route Error
RPL	Routing Protocol for Low power and Lossy Networks
RREP	Route Response
RREQ	Route Request
RTS	Request To Send
S2G	Satellite-to-Ground
S2N	Signal to Noise
S2S	Satellite-to-Satellite
SASNET	SDR-based Ad hoc Space Network
SC	Sequence Number
SDR	Software Defined Radio
SECM	Shanghai Engineering Center for Microsatellites
SMB	Small to Medium Business
SN	Sequence Number
SOCON	Sustained Ocean Observation from Nanosatellites
SOH	State Of Health

SPT	Shortest Path Tree
TDD	Time Division Duplex
TDMA	Time Division Multiple Access
TORA	Temporally Ordered Routing Algorithm
TROPICS	Time-Resolved Observations of Precipitation structure and storm Intensity with a Constellation of Smallsats
TW-1	TianWang-1
UDP	User Datagram Protocol
UHF	Ultra-High Frequency
USU	Utah State University
VANET	Vehicular Ad-Hoc Network
WLAN	Wireless Local Area Network
WSN	Wireless Sensor Network
WSN-ME	Wireless Sensor Network with Mobile Elements

Chapter 1: Introduction

Access to Low Earth Orbit (160 – 2,000km) (LEO) has typically been restricted to military, government and large corporate institutions [1]. Over the past decade, two factors have disrupted this status quo and opened access to LEO for academic intuitions and SMBs alike. The first factor is the advent of the "private space race". Greater competition within the private space industry has caused a dramatic drop in the cost of launching one kilogram into LEO i.e. the "unit cost to LEO". In 2001, NASA's Space Transport System's space shuttle provided a unit cost to LEO, with a fully loaded cargo bay, of approximately $60,000. Thanks in large part to the competitive prices of SpaceX, the minimum unit cost to LEO in 2017 is in the region of $4,000 [2]. Analysis of launch vehicles currently under development has led to predictions of further drops in this cost as a result of increased launch vehicle reusability [3].

The second, and perhaps most influential factor influencing affordable access to LEO, is the introduction of new commonly accepted small satellites classes. This work focuses on the capabilities and applications of CubeSats which, almost always, fall into the Nanosatellite (NanoSat) class. NanoSats have a wet mass of between 1kg and 10kg. Wet mass refers to the mass of the satellite along with the mass of the propellant required to 'lift' the satellite to its desired orbit. Like almost all satellites, the form factors of CubeSats are tailored to match the utilized launch vehicle or deployment mechanism. However, unlike many larger class satellites, there is considerable open-sourcing of the design and implementation of CubeSat components [4].

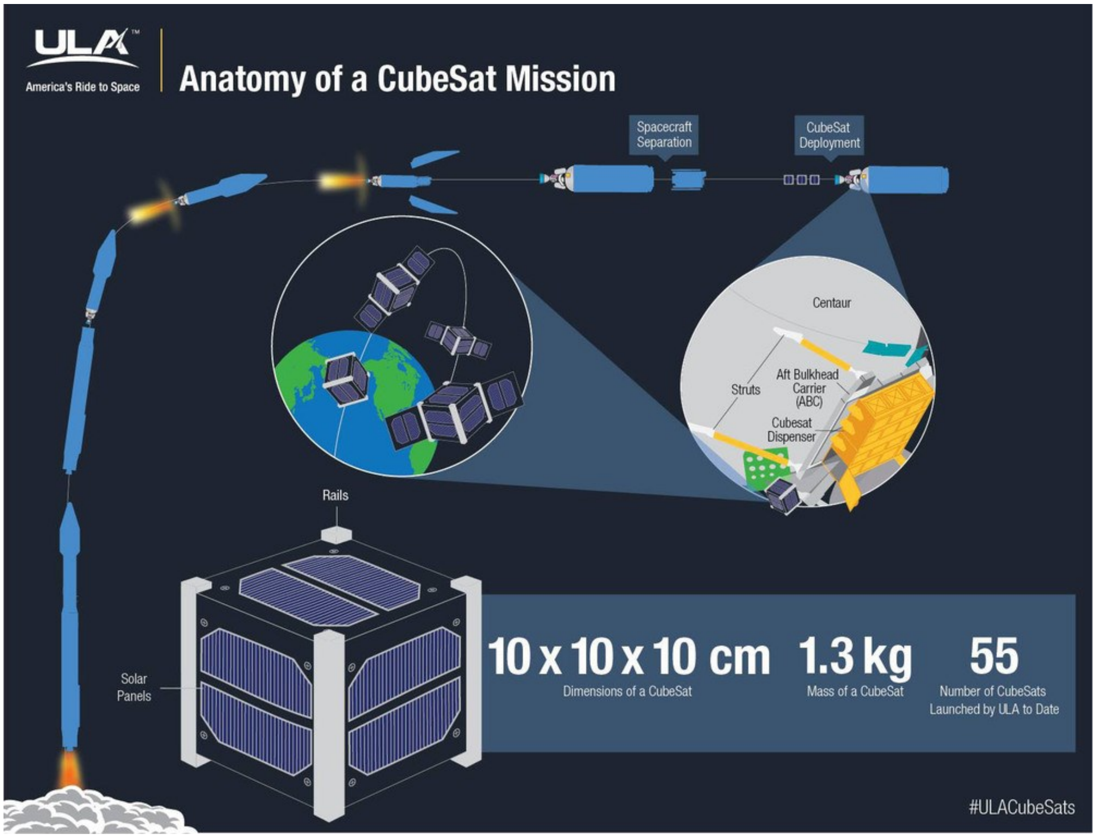


Figure 1. An illustration of the CubeSat form factor and a deployment approach. Here CubeSats are deployed prior to delivery of the primary payload using a dispenser attached to the final stage of the launch vehicle. Image Credit: United Launch Alliance LLC.

CubeSats, as the name suggests, adopt a cube form factor. Each Cube, often referred to as a 'unit', is 10cm to a side (Figure 1). Multiple units are often combined in order to form larger CubeSats. Six unit configurations are typically the largest form factor used [5, 6]. CubeSats are generally constructed primarily of commercial off-the-shelf (COTS) components instead of those designed specifically for the extremes of space environments.

Single unit CubeSats have been shown to be capable of supporting many of the standard sub-systems typically found on larger class satellites which provide: orbital control [7], attitude determination and control (ADCS) [8], communications [9-11], and command and data handling

(C&DH) [12, 13]. Alongside its sub-systems, a CubeSat often carries a small ‘payload’ which may be a scientific instrument or some previously ‘unflown’ component such as an experimental antenna [14]. CubeSats have become increasingly popular within the space industry for testing new technologies and for commercial applications. However, the primary applications for CubeSats remain within educational and academic domains [15].

CubeSats, and other small satellites, have an advantage over larger satellites in their ability to ‘hitch’ a ride alongside primary launch payloads. Primary payloads are designed to match the capabilities of the launch vehicle. Frequently, launch vehicles will have spare volume and lift capacity not required by the primary payload. Multiple CubeSat deployers, or dispensers, have been developed which can make use of this spare volume and lift capacity [16, 17]. Such deployers can often carry multiple CubeSats (Figure 4). In cases where cargo and/or personnel are being delivered to the International Space Station (ISS), CubeSats often hitch a ride. These CubeSats are then launched from the ISS’s dedicated CubeSat deployer.

CubeSat missions have become increasingly ambitious as a result of the reduction of unit costs to LEO and the affordability of COTS CubeSat components [18-20]. This project focuses on a subset of emerging CubeSat missions which involve networked groups of CubeSats; these will be referred to as CubeSat networks (CSNs). Multi-CubeSat missions offer greater redundancy which addresses the platform’s limited power and durability. Missions involving CSNs seek to advance the platform by introducing varying degrees of autonomous cooperation and coordination between CubeSats. It is this cooperation and coordination that presents various new CubeSat mission applications. CSNs have the capacity to enable the collection of greater volumes of scientific data, novel interferometry [21], multi-point sensory data, inexpensive low-bandwidth terrestrial communications and improved air traffic monitoring [22]. The space industry has taken the first crucial steps into designing and testing CSNs with missions such as EDSN (Edison Demonstration of Smallsat Networks) [23] (Figure 2), Nodes (Network & Operation Demonstration Satellites) [13] and Tianwang-1 [24].

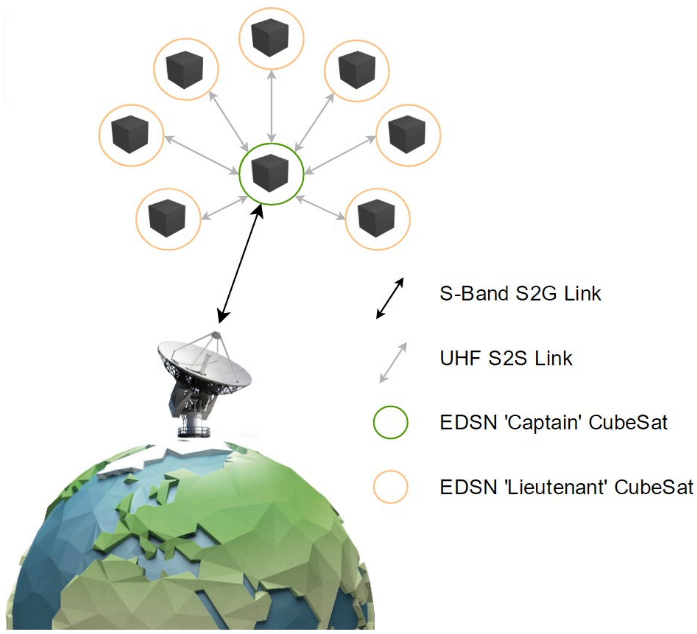


Figure 2. An illustration of EDSN CubeSats in orbit forming a star (hub-and-spoke) topology CSN. Each CubeSat houses two radios, one for S2S the other for S2G communication. Communication to ground is performed only by the current 'Captain' CubeSat.

This work seeks to build upon data from the aforementioned missions, the overall aim being the exploration of the communication approaches employed in CSNs. In particular, this work attempts to identify how CSN based missions may approach communication in order to optimize satellite-to-ground (S2G) data throughput while remaining sensitive to CubeSat energy consumption.

1.1 Objectives

The objective of this work is to explore CSN communication protocol design in the context of the power versus S2G throughput (PvTP) trade-off. This exploration requires an analysis of several fields of research such as; Wireless Sensor Networks (WSNs), Mobile Ad-hoc Networks (MANETs) and CubeSat communications. As there are numerous and varied applications of CSNs, this work narrows the scope of focus to a hypothetical CSN mission.

In the hypothetical mission, each CubeSat is assumed to carry a scientific instrument producing data which must be communicated to ground. The overall performance of the mission's CSN is assessed by the quality, quantity, and/or timeliness of data received at ground. It is assumed that issues relating to data quality are fully addressed. The timeliness of data reception is important in applications such as communications and real-time Earth Observation (EO). The hypothetical mission is assumed to be a generic scientific sensing mission. This work assumes that the value of sensory data is not affected by the time taken for said data to reach ground or, for that matter, by the order in which data packets arrive. Given these assumptions, the CSN's core objective is to maximize the quantity of data received at ground i.e. S2G data throughput.

In terms of energy consumption, S2S communication is generally less expensive than S2G communication. S2S data rates are often in the order of Mbps whereas S2G data rates are frequently as low as 12kbps. This asymmetry differentiates CSN networks from many similar terrestrial networks. Increasing the amount of S2G communication will increase S2G throughput however, it will also consume more energy overall and reduce a mission's lifetime. A reduction in mission lifetime will directly decrease it maximum possible throughput. S2S links may be used to communicate data to a CubeSat which has more available battery capacity and/or a better window of opportunity for S2G communications. However, excessive S2S communication may prove wasteful in cases where all CubeSats have enough battery and suitable S2G communication windows.

This work presents simulations of modified communications protocols developed using the open-source discrete event network simulator OMNeT++ [25]. These protocols were chosen and implemented with consideration to the power versus throughput (PvTP) trade-off for CSNs. Broadly, the optimal approach to the PvTP trade-off is that which requires the least amount of power per Byte of data received at ground. There is potentially no unique solution to the PvTP trade-off. Rather, CubeSat mission designers may chose approaches favouring either throughput or energy consumption based on mission objectives. The PvTP trade-off is affected by numerous mission design choices such as a CubeSat's power and communications capabilities. These capabilities vary from mission to mission and are liable to develop significantly over the coming years.

1.2 Hypothetical Mission

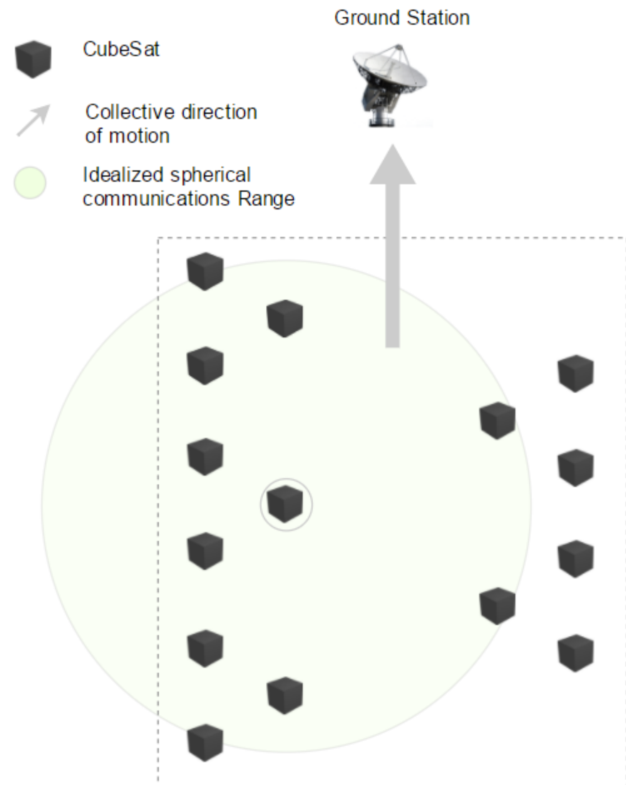


Figure 3. The CubeSats and ground station of the hypothetical as viewed from a higher orbit looking down upon the Earth's surface. The CubeSats are assumed to have an orbital altitude of 550km.

Section 1.1 introduces the general aspects of this work's chosen hypothetical mission. Alongside the PvTP trade-off, the assumptions made regarding the hypothetical mission significantly influence this work's direction. The design of the hypothetical mission is intended to narrow the scope of this work's investigation and reduce the complexity of simulation scenarios. Details regarding the implementation of the hypothetical mission through OMNeT++ are provided in section 4.1.

Notable hypothetical mission assumptions:

- All CubeSats are identical and their components are not liable to failure
- CubeSats do not move relative to one another i.e. they hold a fixed formation (Figure 3)
- The speed, direction and altitude of each CSN ground pass remains constant
- CubeSats are capable of querying the current UTC time as well as their position and velocity from GNSS networks
- Scientific instruments generate data packets of a fixed size (128B) at regular intervals
- The mission's only objective is to return as much science data as possible to ground
- CubeSat radios have ideal spherical communication ranges which are unaffected by the craft's orientation, shadowing or fading (Figure 3)
- Only one ground station is available (Figure 3)

1.3 Thesis Structure

The remaining five chapters of this dissertation are structured as follows. Chapter 2, "State of the Art", examines relevant prior art relating to CubeSat and terrestrial communications as well as detailing the current capabilities and applications of CubeSats. This chapter serves as an extension to the background of the PvTP trade-off and as the rationale behind many of the choices made during protocol and simulation development.

Chapter 3, "Proposed Protocols", presents this work's proposed protocols as informed by the state-of-the-art. In some cases, references are made to restrictions placed upon protocol design due to the practicalities of OMNeT++. These restrictions are discussed further in Chapter 4.

Chapter 3 presents detailed explanations and illustrations of the proposed protocols alongside justifications of relevant design choices.

Chapter 4, "Simulation", deals primarily with the implementation and analysis of this work's proposed protocols simulated using OMNeT++. Several assumptions regarding CubeSat and space-bound wireless communications are discussed. This chapter also details the challenges faced in the implementation and analysis of the proposed protocols.

Chapter 5, "Results", introduces the key metrics chosen for the analysis of the performance of this work's proposed protocols. Results for several simulation scenarios are presented graphically in a number of figures. Each scenario represents a particular parameterization or configuration of the base simulation described in Chapter 4. Discussion is provided for each scenario.

Chapter 6, "Conclusions", presents several areas of discussion and closing thoughts relating to the work reviewed and carried out. The intention of the chapter is to present the findings of this work in the larger context of CubeSats, satellites and the space industry. The chapter concludes by proposing several area of future work.

Chapter 2: State of the Art

The review of literature informing this work is divided into three broad sections: CubeSats, terrestrial communications, and CubeSat communications. The first of these sections provides an in-depth exploration into the CubeSat platform along with the relevant capabilities and applications thereof. This section is followed by an exploration of relevant terrestrial communication technologies with a focus on Wireless Sensor Networks (WSNs) and Mobile Ad-Hoc Networks (MANETs). This final major section seeks to examine, in depth, examples of the latest proposed approaches to CubeSat communications. Several notable secondary areas of research are also discussed in brief. These areas fall outside of the scope of this work but are nonetheless influential in the greater context of space-bound communications.

The "OSI reference model" is referenced extensively throughout this work [26]. The model is used to conceptually separate various aspects of network communications into distinct layers. Four of these layers are of interest in this work. The first, and topmost, "application" layer classifies entities which respond to and generate requests for data from other agents on a network. The bottom three layers are, from the highest down, the network, data link and physical layers. In broad terms, the network layer groups entities which perform packet addressing, sequencing and routing operations. Entities in the data link layer perform error correction and manage access to shared communication media. Entities within the physical layer most commonly handle the conversion of packets into signals which may be received and interpreted by other, connected, network agents. This work proposes two protocols, one MAC protocol within the data link layer and a routing protocol within the network layer.

This following sections cover several areas which provide a fundamental background to CSNs, the PvTP trade-off, and this work's proposed protocols. This chapter is not intended as an exhaustive review of all potentially relevant materials. Rather, this chapter is concerned with works which may clarify the chosen problem, detail potential solutions and inform this work's design decisions.

2.1 CubeSats

CubeSats were first proposed by Dr. Bob Twiggs of Stanford University and Dr. Jordi Puig-Suari of California Polytechnic State University in 1999 [27]. In 2000, the first work detailing a new CubeSat "standard" was published [28]. The CubeSat platform was proposed to address the prohibitive costs and challenges involved in satellite development for academic applications. At the time, there were effectively no standard approaches or components for the design and implementation of small satellites. Researchers relied almost entirely on the installation of instruments, alongside primary payloads, on larger satellites or pursuing the development of dedicated research satellites as lengthy collaborations involving multiple institutions. Frequently, research only required satellites with basic capabilities. These factors created a market for a minimal, low-cost, highly available satellite platform.

In 2003 the first CubeSat was launched on-board a Russian Eurorockot [27]. At the time of writing, May 2017, there have been 487 CubeSats successfully launched or deployed into orbit [29]. Spread across 14 years this number may seem unimpressive, however, approximately 75% of all these launches have taken place prior to 2010. As discussed, this is largely due to the recent boom in the private space industry which has greatly lowered the cost of access to LEO [1].

Thanks in part to a San Francisco based company, named Planet Labs [30], roughly 40% of all CubeSats have been developed by commercial entities. Academic and research institutions have developed approximately 40% and the remaining 20% is divided between civilian and military institutions. In terms of CubeSat applications, roughly 60% of all missions are dedicated to Earth Observation (EO), 20% to technology demonstration, 10% to education. The remaining 10% is divided between various other commercial, military and scientific applications [29].

The core motivation behind the recent popularity of CubeSat missions is their cost. Costs are driven down by three factors, the use of COTS components, open sourcing, and reduced launch expenses. Effectively every component of a modern CubeSat is available in COTS form. Retailers such as Clyde Space Ltd. offer a wide range of products including batteries, radios and attitude determination and control systems [31]. COTS components reduce costs by removing the need to develop, or source, custom components from third parties.

Combined with COTS components, open sourcing lowers costs further by reducing development time and the need for expertize. Open, and often proven, approaches for CubeSat system implementations have become widespread as the platform develops [4]. Although this may seem intuitive, such sharing and open-sourcing of work in the satellite industry has been historically rare.

To date, there is no accepted standards body for the domain. Researchers, such as Dr. Puig-Suari at the California Polytechnic State University (CalPoly), have advanced the domain through the specification several pseudo-standards. Crucially, researchers at CalPoly led the development and design of pseudo-standard CubeSat deployers [32] (Figure 4). A similar pattern can be observed in other CubeSat related areas. For instance, research at the California State University propose a standard CubeSat “satellite bus” design [33]. Recently, “OpenOrbiter”, by Straub et al. from the

University of North Dakota, offers an open pseudo-standard framework for CubeSat development [34].

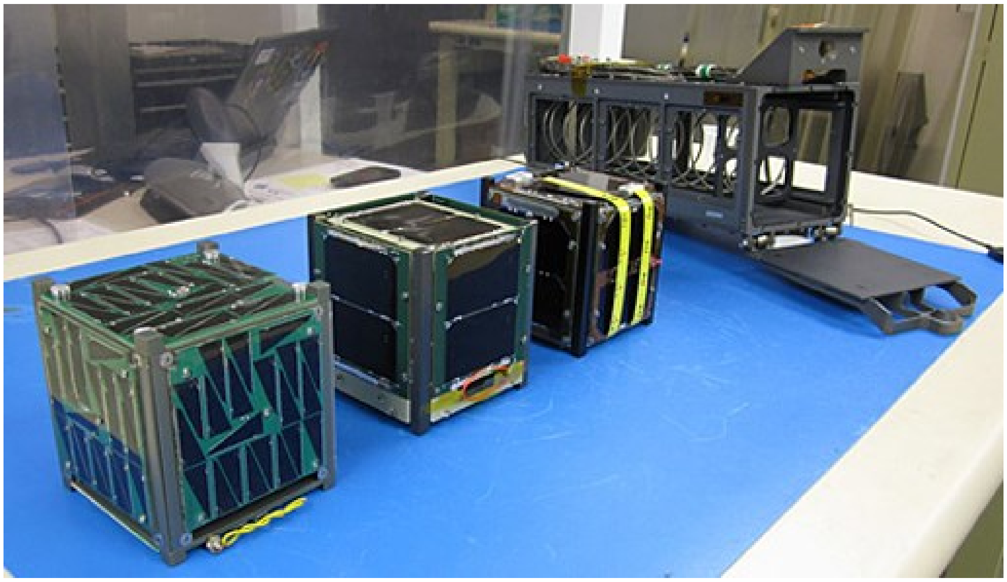


Figure 4. Three 1U CubeSats beside a 3U (Poly Picosatellite Orbital Deployer (P-POD) developed at CalPoly. The spring mechanism used by P-PODs to deploy CubeSats can be seen within the main housing. Image Credit: California Polytechnic State University

Depending on the complexity of a CubeSat, development costs may range anywhere from $50,000 to $250,000 [35]. This can be compared to development costs in the order of millions of dollars for satellites weighing over 100kg. A similar gap has emerged in terms of launch costs. Satellites over 100kg may see launch costs in the order of hundreds of thousands, if not millions, of dollars. This depends heavily on the launch vehicle used and the satellite's orbital requirements (Low, High, Medium, Solar etc.). CubeSats avoid these prohibitive costs by 'hitching' a ride as secondary payloads by using volume and lift capacity not required by primary payloads. With recent developments in multi-CubeSat and CSN missions, multiple CubeSats may take the place of their larger counterparts at a fraction of the cost. CubeSats are also moving beyond LEO as a result of developing deep space and lunar applications [17, 19] (Figure 5).

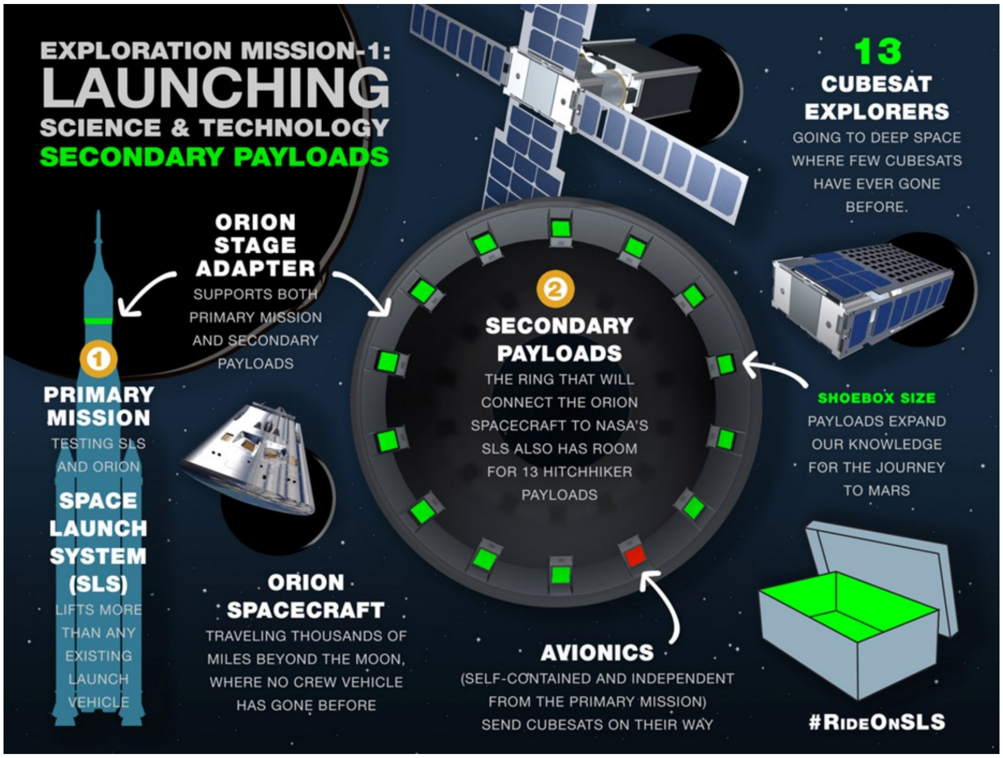


Figure 5. An illustration of the dedicated secondary payload deployers built into the "Orion Stage Adapter" of NASA's upcoming Space Launch System. Existing launch vehicles are generally retroactively fitted with such deployers. SLS is currently projected to launch the first CubeSat into deep space in 2019. Image Credit: NASA.

2.1.1 Capabilities

This section provides an overview of the state-of-the-art technical capabilities of CubeSats. In line with this work's objectives, a focus is placed on technologies relating to communication and power. This section aims to provide context to the upcoming sections on terrestrial and CubeSat communications.

Satellite-to-Ground Communication

There is considerable variance in the implementation of S2G CubeSat communication sub-systems. However, there are broad patterns worth noting. For instance, the most common protocol for S2G communications is AX.25 [36]. CubeSats using AX.25 at the data link layer generally utilize UDP and IP based protocols at higher layers. CubeSat S2G communications sub-systems may consume between 1W to 3W of power during transmission and can achieve data rates between 9.6kbps and 12kbps when using AX.25 [37].

There are some notable outliers to the trends in CubeSat S2G commutations which denote advancements in the domain. In particular, NASA's Dynamic Ionosphere CubeSat Experiment (DICE) mission reports S2G data rates of up to 3Mbps [38]. Such rates were achieved using a custom SDR based sub-system consuming approximately 9W of power and operating within the UHF band. The DICE mission holds the current record for the highest S2G data rate achieved by a CubeSat. At present there are few missions that attempt S2G rates in the order of Mbps, with the notable exception of JPL's ISARA mission [39]. The majority of upcoming missions aim to operate communication rates in the order of hundreds of kbps.

In order to approach protocol design for the PvTP trade-off, baseline state-of-the-art S2G characteristics are chosen. The primary guide for these characteristics is the Tianwang-1 (TW-1) mission [40]. The mission is an ideal candidate for use as a baseline for S2G communication modelling as the mission was designed specifically to test CubeSat networking. TW-1 achieved S2G data rates of 125kbps. Details regarding the energy consumption of the TW-1 S2G sub-systems are unavailable. However, by examining previous and upcoming missions, as well as available work on energy budget analysis, one may assume a peak transmission energy consumption of 3W [41].

Satellite-to-Satellite Communication

The field of CubeSat S2S communication gained popularity following a work published in 2008 on the "Development of a Satellite Sensor Network for Future Space Missions" by Vladimirova et al. CubeSat S2S communication remained purely conceptual until the success of NASA's Nodes mission in 2016 [13].

NASA's Nodes mission and a technology named "Gamalink" [42] inform the current state-of-the-art of S2S CubeSat capabilities. Nodes CubeSats utilized a UHF transceiver and the AX.25 protocol to achieve S2S data rates of 12kbps. As the Nodes mission was intended to demonstrate a number of "firsts" in CubeSat capabilities, the mission designers appear to have opted for a relatively basic approach to S2S communications.

Gamalink is a proprietary SDR based technology developed by Tekever, a Portuguese Aerospace and Defence company. The technology presents a more advanced S2S communication approach than utilized in the Nodes mission. As such, Gamalink may be taken to represent the current state-of-the-art in CubeSat S2S communications. Gamalink has been successfully tested on the TW-1 mission [24]. It is also applied in several other missions such as i-INSPIRE II [43], DelFFi [44] and ESA's Proba 3 [45].

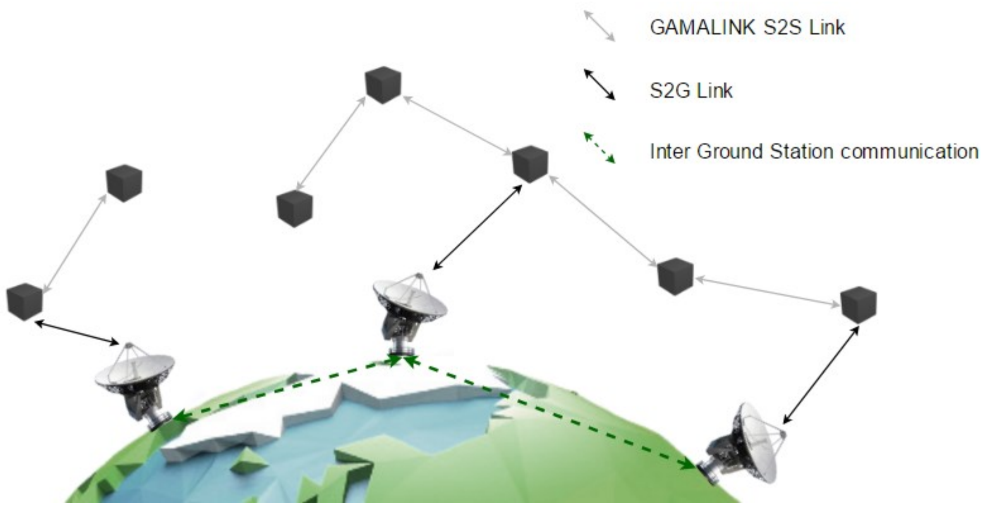


Figure 6. Unlike NASA's EDSN approach (Figure 2), Gamalink seeks to establish multi-hop CubeSat networks capable of communicating with multiple ground stations. Gamalink designers refer to such networks as GAMANETs.

Due to Gamalink's proprietary nature and its potential military applications, details regarding Gamalink are sparse. No openly available information regarding Gamalink's communication protocol use was identified during this work. Tekever make several references to MANETs in Gamalink promotional material, stating that Gamalink implements an "SDR-based Ad hoc Space Network" (SASNET). This is an indication that the state-of-the-art in the field of MANETs is integral to the design and development of Gamalink.

Despite the lack of protocol information, several key data points regarding Gamalink are available. According to promotional material presented to ESA, Gamalink is capable of achieving data rates up to 2Mbps. However, i-INSPIRE mission designers state that Gamalink's maximum data rate is 1Mbps [43]. Gamalink's S2S radio operates in the S-Band, between 2.40 and 2.45GHz, with a bandwidth of 40Mhz. Gamalink consumes a peak of 1.5W while transmitting and up to 200mW while receiving [44]. These details and the assumption of a maximum data rate of 2Mbps inform the configuration of this work's simulation of CubeSat S2S communication.

Battery and Recharge Capabilities

CubeSat energy storage and recharge capabilities vary considerably from mission to mission. The form factor employed for a given CubeSat determines the maximum volume available to house batteries and the maximum surface area available for solar arrays. Folding solar arrays are common place on larger spacecraft. It follows that folding solar panels have also been proposed for use on CubeSat missions [46]. NASA's EDSN mission is used as the example case of the current state-of-the-art for CubeSat energy storage and recharging capabilities.

Each 1.5U EDSN craft carries four lithium ion batteries which combine to provide a maximum energy capacity of 5.2 Amp hours. The craft's bus operates at approximately 8 Volts. This implies a total energy provision of 41.6 Watt hours. Six solar panels provide an average recharge of 1 Watt. A single orbit at a LEO altitude of 500km lasts ~95 minutes. Depending on orbital parameters, each craft will receive varying durations of sunlight during each orbit. Assuming an orbit which is inclined 90 degrees to the Earth's terminator, a CubeSat will be in sunlight for 50% of each orbit (~47.5 minutes). Given these assumptions an EDSN CubeSat may receive approximately 0.79 Watts of recharge per orbit. These characteristics provide context to the energy consumption simulation results detailed and discussed in chapters 5 and 6 respectively.

Other Capabilities

Although the most relevant CubeSat capabilities have been covered in the preceding sections, there are certain other capabilities worth noting. In general, the capabilities of CubeSats have progressed closer to those of larger satellites. Despite strict power, weight and size constraints effectively all major large satellite sub-systems have a corresponding CubeSat equivalent.

Attitude determination and control sub-systems (ADCS) are implemented to ensure appropriate spacecraft orientation. Such systems are critical to ensuring correctly positioned solar panels, antennae and/or payload instruments. In almost all cases, CubeSats must 'de-tumble' prior to deployment. There are numerous tested examples of ADCS technologies for CubeSats [8, 47, 48] several of which are available COTS (Figure 7). Along with ADCS, some basic orbital control and manoeuvrability systems have also been tested at the CubeSat scale [7]. Such systems provide CubeSats with the basic capabilities required to maintain regular orbits and formations.

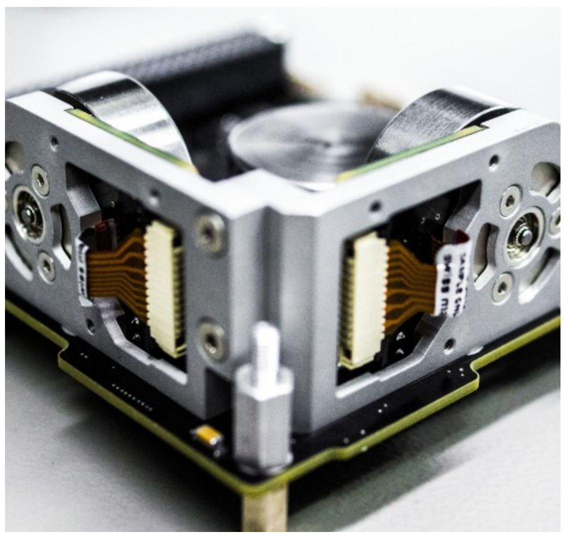


Figure 7. A COTS CubeSat attitude control unit. The rotational velocity of the three reaction wheels shown can be altered to adjust a craft's attitude. Image Credit: Clyde Space Ltd, All Rights Reserved.

Through communication with Global Navigation Satellite Systems (GNSS) a CubeSat may acquire precise time, velocity and positional information. Missions often require CubeSats to periodically update such information in order to coordinate in-orbit operations and S2G communications. One work by Glennon et al. on CubeSat time synchronization provides a clear overview of potentially beneficial applications thereof within multi-CubeSat missions [49].

Finally, it is worth noting that Gamalink provides functionality beyond that of S2S communication. Gamalink also provides the secondary functions of GNSS receiving, attitude determination, ranging (5m resolution) and distributed clock synchronization.

2.1.2 Applications

This section examines a number of CubeSat missions. Two categories of application are considered; sensing missions and CSN missions. This is not intended to assert that sensing missions and CSN missions are disjoint. In fact, CSN missions are highly suited to collaborative sensing applications.

Sensing Missions

When approaching the CSN PvTP trade-off, it is beneficial to establish a broad application case. As discussed, Earth observation is the most popular application of CubeSats to date. More generally, the majority of CubeSat missions have involved, to varying extents, some form of sensing. The hypothetical mission chosen by this work may be considered a simplified case of a CSN sensing mission.

Two recent sensing missions are worth detailing in the context of CSNs: 3Cat-2 [50] and RAVAN (Radiometer Assessment using Vertically Aligned Nanotubes) [51]. 3Cat-2 involves a 6U CubeSat developed at the Universidad Politécnica de Cataluña. The mission launched in August of 2016 on-board a CZ-2D (Chang Zheng-2D) operated by the CNSA. 3Cat-2's S2G downlink operates at a maximum of 115kps. This is a similar data rate as achieved by the Tianwang-1 mission which informs this work's simulation of S2G communication.

3Cat-2's particular application case is ocean altimetry by means of GNSS-Reflectometry. 3Cat-2 performs altitude observations by examining the scattering and reflection of GNSS based signals off bodies of water. These are an 'active' form of measurement which depend on incident signals; Radar is another example of active measurement.

Mission developers of 3Cat-2 have not stated a direct desire to pursue a future multi-CubeSat mission. However, 3Cat-2's active sensing is uniquely suited to adaption with a CSN. Coordinated and synchronized measurement of signals by multiple craft in orbit could greatly improve observation fidelity and provide unique multi-dimensional data. Comparatively, 'passive' EO, such as direct imaging, benefits less from adaptation with a CSN.

RAVAN is a 3U CubeSat developed at the Johns Hopkins Applied Physics Laboratory. RAVAN was launched in November of 2016 aboard an Atlas-5 as part of NASA's ELaNa (Educational Launch of Nanosatellites) program. RAVAN mission designers clearly specify future intentions to develop a constellation of RAVAN craft. In satellite nomenclature, a constellation is considered to be a formation of satellites evenly distributed over the surface of the Earth.

RAVAN carries an experimental carbon nanotube based radiometer. RAVAN's instrument performs multi-spectral measurements of outgoing radiation from Earth's surface. These measurements reveal trends regarding Earth's Radiation Budget (ERI) which are valuable to climate scientists. As mentioned, RAVAN is intended as a first test in a larger plan to develop a constellation of craft [52] (Figure 8). The spacing of the forty proposed RAVAN craft prohibits S2S communication using current technologies. Nonetheless, RAVAN is strong example of the growing interest in multi-CubeSat missions.

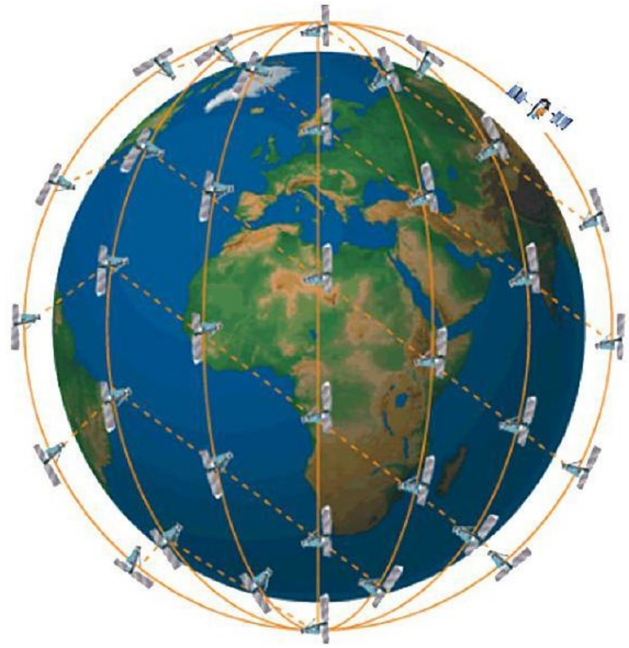


Figure 8. A conceptual illustration of the proposed RAVAN constellation. Image Credit: John Hopkins University Applied Physics Laboratory.

The examples of RAVAN and 3Cat-2 are in no way intended to illustrate a comprehensive study of CubeSat sensing application. Other notable sensing application include: CeREs (a Compact Radiation belt Explorer) [53], LAICE (Lower Atmosphere/Ionosphere Coupling Experiment) [54], and SOCON (Sustained Ocean Observation from Nanosatellites) [55]. Such applications have created interest in the development of CSN enabled multi-point measurement, in-orbit interferometry [56] and synchronized observation. In this regard, CSNs represent an obvious next step in the advancement of CubeSat sensing applications.

CubeSat Network Missions

There are three missions of note in the area of CSNs: EDSN, Nodes, and Tianwang-1 (TW-1). Of these missions, both Nodes and TW-1 have successfully flown. NASA's eight CubeSat "Edison Demonstration of Smallsat Networks" (EDSN) mission was lost due to a failure during launch.

The mission is still worth examining however as two of the remaining EDSN craft were used during the successful Nodes missions.

Two articles detailing the EDSN mission were published in 2014 prior to the loss of the mission in November 2015. The first work examines the inter-satellite communications architecture of the mission [57]. The second work details lessons learned during development [58]. The primary objective of the EDSN mission was to implement the autonomous communication and coordination of CubeSats.

Each ESDN craft is a 1.5U CubeSat weighing ~1.73kg. A modified Samsung® smartphone provides activity scheduling and execution for each craft. Several secondary COTS micro-controllers handle the CubeSat's various activities which include; GNSS communication, C&DH, scientific measurement and ADCS. Each EDSN craft's scientific payload is an instrument designed to characterize radiation in LEO called the "Energetic Particle Integrating Space Environment Monitor" (EPISEM). Although the scientific objectives of the mission were secondary to the implementation of a CubeSat network, EDSN falls within the category of sensing applications. EDSN's sensing objectives are not entirely dissimilar from those of the RAVAN mission.

In terms of communications and power capabilities, the works published on EDSN provide a wealth of information. Each craft houses three primary radios: A MicroHard MHX2420 transceiver for S-Band S2G communication, an AstroDev Lithium 1 UHF transceiver for S2S communication and a StenSat UHF transmitter for beaconing (Satellite beaconing is required by the North American Aerospace Defence Command (NORAD)). The AstroDev transceiver enables S2S communications at 9.6kbps using the AX.25 protocol at the data link layer. The Nodes mission scaled this data rate back to 1.2kbps, potentially to increase maximum S2S communication range. Details regarding the MHX2420's S2G data rate capabilities are not provided.

As mentioned, EDSN employs an AX.25 link layer for S2S communication. For the majority of CubeSat missions, AX.25 along with a basic application layer, the entities of which communicate directly with the link layer, is sufficient [37]. However, S2S communication introduces new challenges which warrant more involved approaches. In the case of EDSN, a custom "Captain – Lieutenant" (Cpt/Lt) protocol was designed on top of AX.25.

The network formed by the EDSN craft is referred to as a "hub-and-spoke" (or star) network (Figure 2). One craft is designated as the "Captain" (Cpt) and all others are designated as "Lieutenants" (Lts). In general terms, the Cpt acts as a central router to ground. All Lts send their data exclusively to the Cpt (Figure 9). The Cpt then communicates as much of this data to ground as possible. Lt communication in EDSN is controlled solely by the current Cpt. The Cpt sends six ping messages over 50s seconds. Each set of pings specifies one Lt from which the Cpt is requesting data. Only after receiving a valid ping does a Lt forward its data to the Cpt. This scheme, of Cpt request followed by Lt response, ensures no overlapping communications can occur on the shared S2S frequency.

There is no acknowledgment scheme employed in EDSN's Cpt/Lt protocol. Lts send one "state-of-health" (SOH) packet followed by all queued science data packets. The Cpt prioritizes the communication of these SOH packets to ground and treats science packets generated by Lts, or by its own instrument, in a FIFO manner. After ending a communication session with a Lt, the Cpt will proceed to ping each remaining Lt in a fixed order (Figure 9). The Cpt will wait up to four minutes for a response from a pinged Lt before moving on to ping the next Lt.

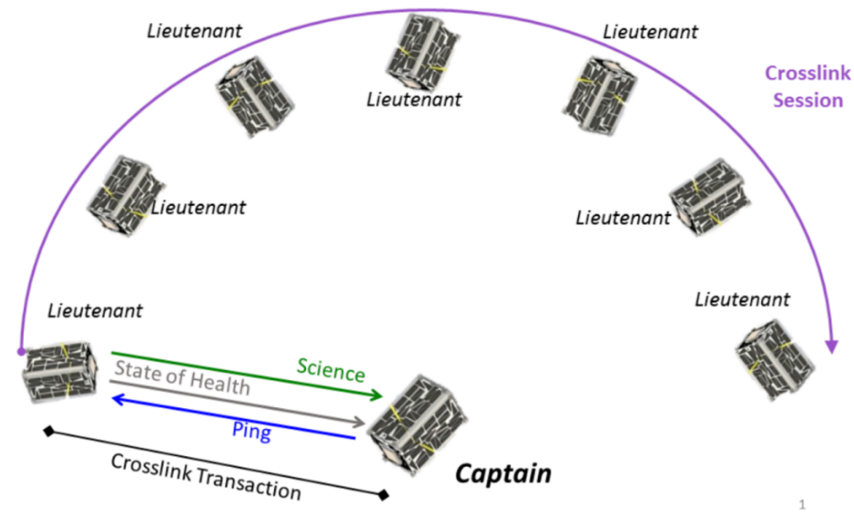


Figure 9. The Cpt/Lt protocol. EDSN designers refer to S2S communication as crosslinking. The Captain pings a Lieutenant before receiving state-of-health and science data packets. Image Credit: NASA Ames Research Centre

The Cpt role is "rotated" amongst the EDSN craft in a pre-defined fixed pattern. There is no real-time logic or election employed. Each craft periodically receives GNSS time in order to determine whether to assume the role of Cpt. If a craft cannot get GNSS time it does not participate in either S2S or S2G communications. The duration for which a single craft holds the Cpt role is referred to as a "minor cycle". Each minor cycle lasts roughly 25 hours and includes three to four S2S sessions. Each session involves a single attempt by the Cpt to communicate with each Lt in the network.

The Cpt self-determines one, and only one, S2G session during its minor cycle. Such sessions are scheduled by predicting the next ground station fly-over period (window) using the craft's GNSS location and velocity. As S2G communication occurs using a separate radio at a separate frequency to S2S communication, S2G sessions can take place in parallel with S2S sessions. Eight minor cycles, one for each craft, form a major cycle. EDSN mission planners predicted that after three major cycles (three and one half weeks) EDSN craft would have drifted too far apart (>120km) for S2S communication to be feasible.

During a minor cycle it is not feasible for all Lts to keep their S2S transceivers continually on and in receiver mode. As such, along with the Cpt pinging scheme, all S2S communication within a minor cycle follows a fixed schedule. Following this schedule a Lt will begin listening for Cpt pings at a predetermined time during each minor cycle.

During a minor cycle each craft updates it's GNSS time, position and velocity only once. EDSN mission designers predict the maximum relative clock drift between craft to be 12 seconds. As such, a Lt will begin listening for Cpt pings 30 seconds before its scheduled S2S session. The Lt will also continue listening 30 seconds after the expected sixth and final ping. S2S session start times within each minor cycle are determined by a table of offsets. These offsets are relative to the start times of each minor cycle. Each craft is pre-programmed with the same minor-cycle start times and offset tables.

Following directly from the work on EDSN, the Nodes mission was successfully deployed from the ISS in May of 2016. Nodes employed two leftover CubeSat's from the EDSN mission which were used during the testing and development of EDSN. Many of the aforementioned salient aspects of the EDSN mission remain. Despite only involving two craft, Nodes was able to achieve many of the objectives of EDSN. The changes by the Nodes mission are detailed in a work published in 2016 following the mission's successful launch, deployment and conclusion [13].

Where EDSN focussed purely on S2G communication, Nodes advances one step further by introducing to demonstration of Ground-to-Satellite (G2S) remote commands. In Nodes, an objective was set to communicate a command from ground to the elected Cpt. The Cpt would then forward this command to the Lt for execution. Unlike science and state-of-health packets, command packets are implemented along with specialized command acknowledgement and response packets. Although this work focuses purely on S2G communications, it is worth noting that Nodes was the first demonstration of the indirect command and control of CubeSats using CubeSat S2S communication.

Nodes introduces several notable changes to the Cpt/Lt protocol. Firstly, 12 pings over 110 seconds are utilized rather than 6 pings over 50 seconds. In place of EDNS's a fixed minor-cycle schedule Nodes craft dynamically negotiate between the assignment of the Cpt role. During this negotiation, a default Cpt craft compares metrics relating to battery voltage, the amount of science data collected and the predicated duration of the next ground station fly over.

At the start of each minor cycle the selected default Cpt initiates captaincy negotiation (Figure 10). Once the default Cpt has compared its own metrics with that of the Lt, it will either continue as Cpt or send a "promote" command to the Lt. The default Cpt will only vacate the Cpt role following the receipt of a promotion acknowledgement from the new Cpt. In general, the Cpt will continue operations regardless of the presence of any communication with the Lt. All other aspects of the Cpt/Lt protocol are unchanged.

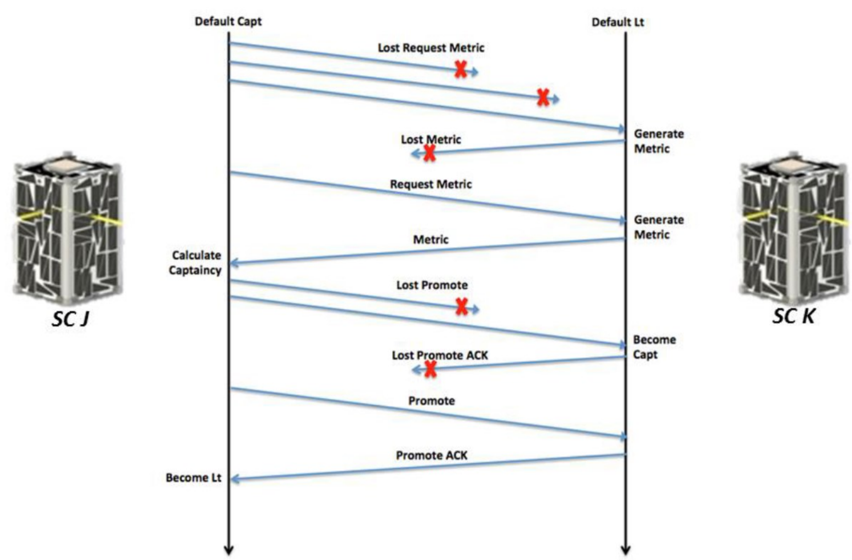


Figure 10. A timeline of the Captaincy negotiation process carried out between the two Nodes spacecraft (SC). Image Credit: NASA Ames Research Centre

Over the course of its three-week mission, Nodes completed and/or exceeded all five of its mission objectives:

- The collection and receipt of five 'sets' of science data
- Five successful S2G sessions
- One successfully executed indirect command
- Two successful captaincy negotiations
- The collection and receipt of 20 days of spacecraft state-of-health data

Of the total 470 science packets generated (size undisclosed) a total of 356 were successfully received at ground representing ~25% packet loss. Five successful negotiations were carried out and 165 commands were executed by Nodes craft.

Nodes mission designers lay out numerous desirable enhancements such as: improved clock synchronization, inter-sat ranging, multi-hop routing, further acknowledge systems, delay tolerant networking principles, multiple ground stations and the interlinking of multiple Cpts to form "clusters of clusters". Several of these suggested areas of future work are partially addressed in the protocols proposed and simulated in this work.

The Tianwang-1 (TW-1) mission, also referred to as STU-2, was a three CubeSat CSN mission involving numerous commercial and academic organizations led by the Shanghai Engineering Centre for Microsatellites (SECM). The majority of the published work relating to TW-1 details its ADCS and propulsion systems [47, 59]. A presentation by Wu et al. during the 30th Annual AIAA/USU Conference on Small Satellites in 2016 offered a brief overview of the mission's Gamalink communication system [40]. However, as discussed, publicly available details of the proprietary Gamalink technology are sparse. For known information on Gamalink refer to section 2.1.1.

TW-1 was launched in late September of 2016, three months after the deployment from the ISS of the Nodes mission. TW-1 was, like Nodes, primarily a technology demonstration mission. TW-1's objectives were to flight test Gamalink, an ADCS and a propulsion (orbital control) system. TW-1 consisted of one 3U CubeSat (TW-1A) and two 2U satellites (TW-1B & TW-1C). TW-1A housed the mission's experimental propulsion systems. This propulsion systems allowed TW-1A to remain within S2S communication range of TW-1B for a longer period than would have been possible without orbital control. TW-1 collected data on aircraft flight patterns using an on-board Automatic Dependent Surveillance (ADS) receiver. TW-1 also performed earth observation using visual spectrum cameras to image terrestrial polar regions. In line with its objectives, TW-1 also carried out several in-orbit tests on its experimental sub-systems. S2G communication of the mission's collected data was demonstrated at a rate of 125kbps. Apart from this figure, there is no further relevant information available regarding TW-1's power or communication capabilities.

There are several other missions, besides EDSN, Nodes, and TW-1, that may provide insight into the state-of-the-art of CSNs such as: ESA's AIM COPINS [60], GomX-4 [61, 62] and Proba-3 [45], NASA's CPOD [63] and TROPICS [64], QB50 [65] and OLFAR [56]. With the exception of COPINS, which was defunded, these missions are, as of May 2017, in development or awaiting launch.

2.2 Terrestrial Communications

The design of CSN communication protocols may be guided by work from several fields of terrestrial communications research. In this section two such fields are explored. These fields are Wireless Sensor Networks (WSNs) and Mobile Ad-Hoc Networks (MANETs). These fields have considerable breadth and depth, as such a focus is placed on survey and review style publications.

The following sections on WSNs and MANETs attempt to identify the most relevant sub-domains within each field. As the majority of CubeSat applications involve sensing operations, parallels exist between CSNs and terrestrial WSNs. Within the field of WSNs works relating to data collection and energy conservation are considered most relevant to this work.

MANET related works are relevant in their treatment of the mobility of network members. Particular attention is paid to Flying Ad Hoc Networks (FANETs). Like WSNs, FANETs share many of the same properties as CSNs. FANETs are expected to experience intermittent, potentially predictable, access to a greater and more 'static' network. In the case of CSNs, this static network is represented by one or more ground stations.

2.2.1 Wireless Sensor Networks

Common communication challenges within WSNs relate to the unpredictable failure and resource constraints of network nodes. These challenges impact, to varying degrees, on a WSN's ability to perform data collection or data dissemination. A sink (collection) or originator (dissemination) of data in the context of CSNs is an Earth based ground station. This work is concerned with energy efficient data collection (the PvTP trade-off) with the added complication of node mobility.

A survey by Rault et al. focuses on energy efficiency in WSNs [66]. The authors approach the domain by examining several areas of WSN application such as healthcare, transportation and industry. For each area the authors outline WSN application requirements such as scalability, mobility, and security. Table 1 extends this assessment with two application cases relevant to CSNs.

Table 1. An extension of Rault et al.'s table presented in [66]. The extension includes two relevant CSN applications. ET: Extra-Terrestrial.

		Scalability	Coverage	RT Delay	QoS	Security	Mobility	Robustness
Space	CSNs – Earth Observation	-	+	--	+	-	++	+
	CSNs – ET Science	--	-	--	+	-	+	+

Rault et al. detail several aspects of low-power WSNs and the trade-offs relating to approaches to increase energy efficiency. Several low power WSN standards are explored such as IEEE 802.15.4 [67], Bluetooth Low Energy (BLE) [68], and the IPv6 based "Routing Protocol for Low power and lossy networks" (RPL) [69]. No one standard presents itself as being an obvious choice for the CSN PvTP trade-off. However, clear benefits can be seen in elements of RPL and ISA100.11a [70], one of the many extensions of 802.15.4.

Beyond low power WSN standards, several core WSN approaches are highlighted by Rault et al. Although these approaches are relevant in some regard to CSNs, the areas of highest relevancy to this work are: duty cycling schemes, cluster architectures, energy as a routing metric and sink mobility. Each of these areas are explored, to some extent, by this work's simulation scenarios.

Rault et al. discuss a number of approaches to the trade-offs involved in the implementation of energy efficient WSNs. Three techniques are discussed: Multimetric protocols, cross layer optimization (CLO), and Multi-objective optimisation (MOO). Of the three techniques, CLO bears the most relevance to CSNs. The authors define CLO as "solutions exploit(ing) interactions between different layers to optimise network performances" (Figure 11). This work adopts Rault et al.'s definition of CLO. Two particular surveys are cited as being authoritative on the WSN CLO domain [71, 72]. Several examples of CLO are provided in Figure 11.

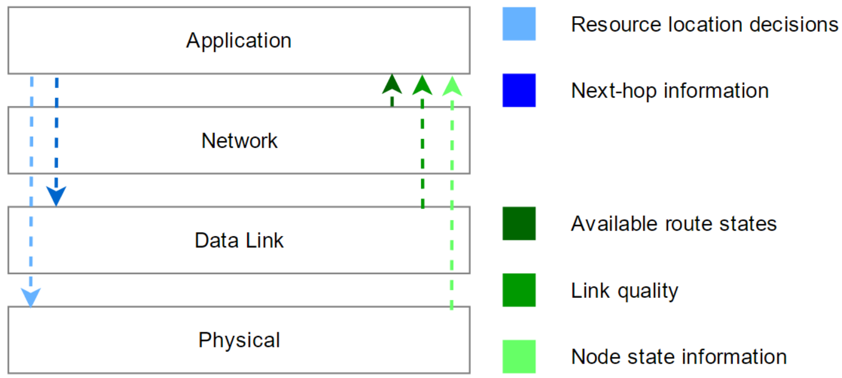


Figure 11. Examples of information that may be passed between layers. This is a departure for the treatment of layers as independent black boxes.

No one approach or standard is singled out as the obvious choice for energy efficient WSNs. Rather, Rault et al. point to the importance of adapting approaches to applications. The authors state clearly that regardless of the application or approach, the introduction of greater CLO is expected to advance the WSN field. CLO is a common and favoured theme throughout recent literature relating to WSNs and MANETs. The exact definition of CLO differs somewhat depending on the author, application and domain.

WSN data collection is examined by Francesco et al. in a survey paper published in 2011 [73]. The work is particularly useful as it focuses on WSNs with Mobile Elements (WSN-MEs). The authors concentrate on mobility while maintaining and referencing the existing relevant state-of-the-art in WSN routing and energy management. Francesco et al. describe a number of WSN mobility scenarios the most relevant of which is the “Mobile Peer” scenario (Figure 12). With CSNs, one may consider the ground station as moving into range of a satellite or a satellite moving over ground (Figure 3).

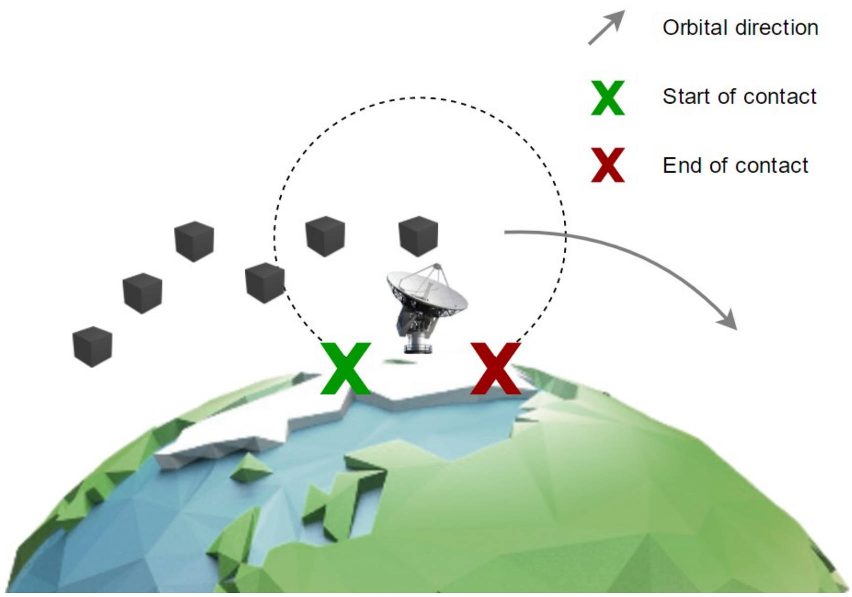


Figure 12. A mobile peer architecture is similar in several regards to orbiting CubeSats passing over a ground station due to intermittent contact windows.

Data collection is divided by Francesco et al. into three distinct phases: 'Discovery', 'Routing', and 'Data Transfer'. In the discovery phase, network members (nodes) attempt to identify their contactable neighbours. During the routing phase, nodes attempt to establish routes to unreachable nodes through identified neighbours. Finally, given an established route or data destined for a neighbour, a node can begin the communication of data which constitutes the data transfer phase.

Francesco et al. state that the treatment of the discovery phase is critical to the performance of WSN-MEs. The authors highlight issues with discovery protocols which rely on forming a schedule for discovery attempts. The timeliness of the discovery carried out by a mobile element determines maximum windows for communication. Schemes based on periodic or scheduled discovery attempts are liable to suffer from reduced communication windows due to ill-timed discovery. For instance, in the case of CSNs, ill-timed discovery attempts could lead to CubeSats discovering that the ground station is within range as they finish their ground passes (Figure 12).

Francesco et al. highlight a discovery approach that utilizes low-power short-range radios to asynchronously waken nearby nodes prior to data communication using a longer range higher power radio [74]. Other methods involving machine learning are noted for their potential to avoid the pitfalls of periodic/scheduled discovery. Such methods may, with or without prior heuristics or topology knowledge, converge on optimal or adaptive discovery schedules over time [75]. In the case of CSNs, CubeSats can determine their next communication window with ground using GNSS time, positon and velocity information. The challenges of the discovery of S2S neighbours are similar to those of WSN mobile element discovery.

The data transfer phase, which follows the discovery phase, is primarily concerned with communication quality and MAC schemes. The authors state that WSN-ME data transfer is a field that requires further work. The authors note a stop-and-wait protocol [76] as well as an automatic repeat request (ARQ) scheme [77]. Further references to specific well-established MAC schemes are sparse. Francesco et al. make a clear recommendation that network coding schemes require greater attention in relation to WSN-ME data transfer [78]. MANETs and WSN-ME bear numerous similarities. Work on data transfer within the domain of MANETs tends to be broadly applicable to WSN-MEs. Why Francesco et al. avoid the direct evaluation of relevant MANET related work is unclear. It is possible the authors intend to stress the importance of adapting MANET approaches specifically for WSN-MEs.

The WSN routing phase, as highlighted by Francesco et al., is considerably more developed than the discovery or data transfer phases. The authors assume that the motion of network elements is not controlled or periodic. In this regard, the works discussed may be relevant to the S2S communication of CubeSats lacking orbital control capabilities. Routing approaches for uncontrolled mobile WSN elements are classified by the authors into "flat" routing and "proxy-based" routing. In flat routing schemes, all nodes behave in the same fashion. In proxy-based schemes, certain nodes may take on additional routing or proxy roles.

Several approaches to routing are discussed by Francesco et al. Three are worth noting in brief: A modified Optimized Link State Protocol (OLSR+) [79], Energy-Aware Routing to Mobile gateway (EARM) [80], and a cluster based approach by Somasundara et al. [81]. OLSR+ adapts to the mobility of element by sharing velocity information between nodes. With this added information nodes can estimate the future link stability to avoid unnecessary abandonment or predict route switches. EARM implements an adaptive power approach wherein nodes may boost their transmission power to a mobile node as the node moves out of range. EARM's core logic allows nodes to decide the point at which a direct link should be abandoned in favour of a multi-hop route. Somasundara et al.'s approach employs adaptive clustering. Nodes collaboratively and dynamically form clusters and elect cluster heads. These heads are elected based on their distance from some mobile sink (Figure 13). Cluster heads act as routers to the mobile sink and manage inter-cluster communication. Somasundara et al.'s approach informs to the cluster based MAC approach developed by this work. Unlike Somasundara et al.'s approach, this work does not employ a cluster based routing strategy such as the Cluster Based Routing Protocol (CBRP) [82]. Rather, this work adapts a routing protocol known as the "Dynamic MANET On-demand routing protocol" (DYMO) [83] for use with a cluster based MAC protocol.

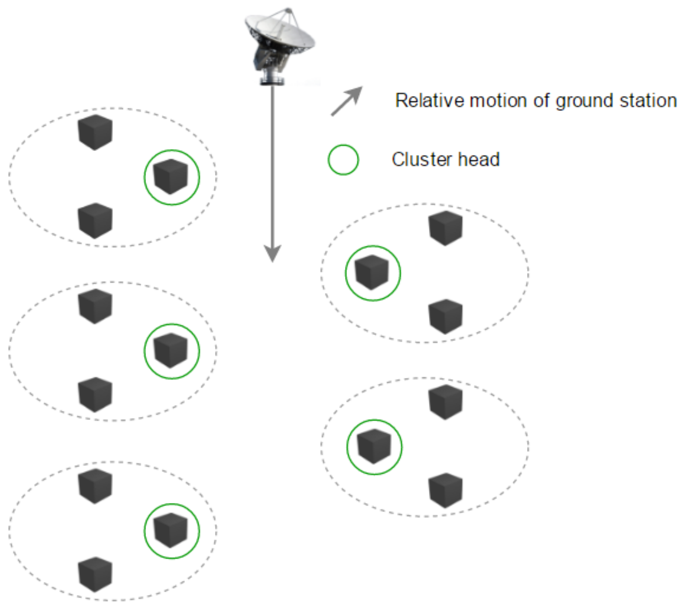


Figure 13. An example of network cluster forming and election of cluster heads. With the introduction of a mobile sink (ground station), clustering can be adapted to ensure the election of cluster heads with the longest contact durations and/or most resources [81].

Finally, to compliment the discussed WSN survey materials a more targeted work is considered. "Efficient data collection in wireless sensor networks with path-constrained mobile sinks" by Gao et al. approaches the PvTP trade-off directly through the design of a novel routing protocol [84]. Increasing the relevancy of this work to the CSN domain is the focus by Gao et al. on WSN's with mobile sinks. Each mobile sink can be considered analogous to a CSN ground station. The proposed protocol is given the name "Maximum Amount Shortest Path" (MASP). The authors determine the formal properties of MASP and use OMNeT++ to simulate and analyse its performance.

MASP outperforms a common approach to the determination of the most efficient routes to adopt called Shortest Path Tree (SPT). There are several implementations of SPT, but a common approach is to construct a tree of possible routes to a destination. This tree can then be search with an algorithm such as A* with a heuristic cost function which represents the energy cost of using a given route. MASP outperforms SPT by utilizing a genetic algorithm which solves a multi-dimensional optimization problem. This problem is based on known and computable routes and route heuristic functions.

The work of Gao et al. represents a state-of-the-art routing approach relevant to the PvTP trade-off in WSNs. The use of “energy as a routing metric” is considered critical by Rault et al. in developing energy efficient data collection approaches. MASP develops on this concept of energy as a routing metric and introduces an intricate, yet performant, solution. However, MASP can only determine efficient routes given successful and timely neighbour discovery. Without the introduction of an appropriate discovery scheme MASP’s performance is fundamentally limited. This lack of development in the area of discovery is further discussed by Francesco et al.

2.2.2 Mobile Ad-Hoc Networks

MANETs bear obvious similarities to CSNs. MANET research tends to be less concerned with resource and capability constraints placed upon network members. Rather, MANET research tends to focus on an approach’s ability to efficiently and reliability enable a network to self-organize and communicate.

In MANETs no initial shared knowledge of the network is assumed. With CSNs, although several CubeSats may be deployed together, countless factors could cause unpredictable failures or orbit perturbations. LEO is an environment of extremes with intermittent flares of radiation, hazardous solar winds, fluctuating magnetic activity and temperatures ranging from -170 to 150 degree Celsius. The MANET technologies which will prove most valuable in the context of CSNs are those which handle topology changes and enable energy efficient communication.

The most discussed and active topic within MANET research is that of routing. MANET routing protocols are generally divided into three primary classes: reactive, proactive and hybrid. Reactive protocols attempt to establish routes only as required whereas proactive protocol attempt to establish routes in-advance of communication. Hybrid protocols implement a mix of reactive and proactive approaches. This is generally achieved by restricting reactive or proactive behaviours to certain areas of a network. The methods by which protocols maintain and discover routes may or may not differ between protocols of these classes.

This section places a focus on reactive protocols. Generally, proactive and hybrid protocols are unsuitable for CSNs. Proactive protocols are typically designed for low mobility networks with reliable links. Hybrid protocols can provide a 'best of both worlds' between reactive and proactive protocols. However, hybrid approaches are typically best suited for larger networks wherein the additional overheads of a complex protocol scales more favourably.

Several relevant protocols are examined by Mohseni et al. in their survey of MANET routing protocols [85]. The conclusions of the authors reinforce the assertion that reactive protocols are better suited for CSNs. The authors state that, comparatively, proactive protocols tend to require more power, bandwidth and incur larger overheads. The primary benefit of proactive protocols is the constant availability of routes which lowers latency and increases the consistency of communication throughout the network.

Mohseni et al. discuss a number of the most well-known reactive protocols such as "Dynamic Source Routing" (DSR) [86], Ad hoc on Demand Distance Vector (AODV) [87], Temporally Ordered Routing Algorithm (TORA) [88] and CBRP [82]. Of these it is worth providing a brief overview of DSR and AODV. In DSR, a node (the originator) broadcasts a route request (RREQ) for a given target node. Non-target nodes add their address to an incoming RREQ packet and rebroadcast it. Loops are avoided by nodes dropping RREQ packets to which nodes have already added their addresses. Once an RREQ reaches its target it contains a list of nodes representing one possible route from originator to source (Figure 14). An attempt is then made to use this route to send a unicast route response (RREP) from the target to the originator. Through this approach, DSR allows nodes to build up multiple routes for a given target. In DSR, due to the construction of routes within packets, route message sizes grow in proportion to the overall network size.

AODV builds upon concepts from DSR and a proactive protocol known as Destination-Sequenced Distance-Vector routing (DSDV) [89]. Unlike DSR, AODV only requires the inclusion of the originator and target addresses within a route packet. AODV implements the route packet sequences number approach of DSDV in order to avoid routing loops and determine the 'freshness' of routes. AODV also introduces features such as 'intermediate-RREPs' and 'Hello' messages. An intermediate-RREP may be generated by an intermediate node in response to incoming RREQs if said node already has a route to RREQ's target. 'Hello' messages may be periodically generated in order to detect broken routes and maintain up to date route costs.

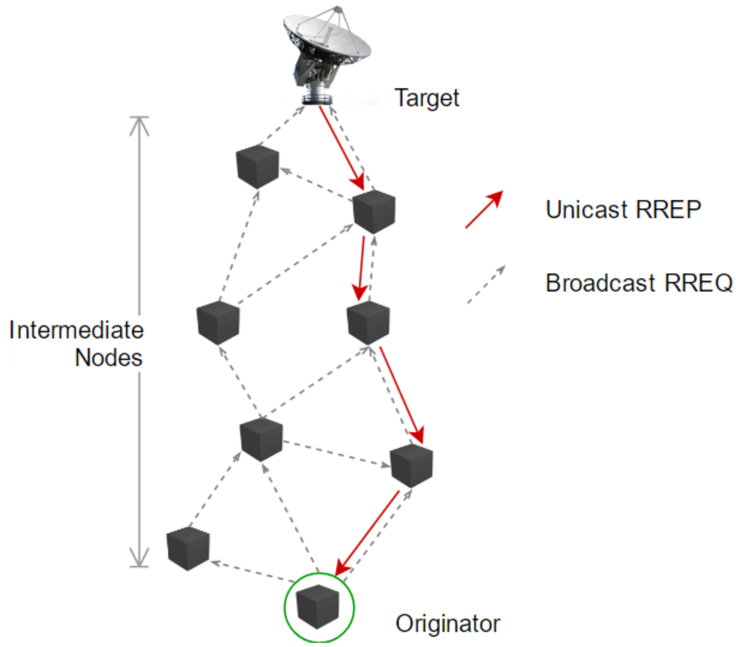


Figure 14. An illustration of the broadcast RREQ unicast RREP approaches used by DSR and AODV. Targets will generate RREPs for each arriving RREQ. RREQs which revisit nodes are dropped. The paths of dropped RREQs are not illustrated above.

The routing protocol utilized by this work, DYMO, is a modification of AODV. Recent IETF specification drafts have begun to refer to DYMO as "AODV version 2". This work's implementation of DYMO within the network simulator OMNeT++ is based on an older IETF specification draft [83] which uses the DYMO naming convention. The primary differences between DYMO and AODV relate to implementation. The core route discovery and maintenance approaches of AODV are largely unchanged by DYMO. Details of DYMO are discussed further in the chapter 3.

Vehicular Ad-hoc Networks (VANETs) are a sub-domain of MANETs. There are notable parallels between CSNs and VANETs such as the separate treatment of vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication. Despite initial similarities, a survey of the protocol stacks in the domain by Mohammad et al. reveals several undesirable facets of VANETs [90]. As mentioned, there is generally less focus in the field of MANETs placed on resource constraints. VANETs follow this trend as the majority of applications focus on planes, trains, and automobiles [91]. Also, security is a central topic in recent VANET research. Although security has relevance to CSNs, communication security is beyond the scope of this work.

Mohammad et al.'s survey shows the strong preference of VANET protocol stacks towards mobile IPv6 capable technologies and flavours of the 802.11 protocol stack. Implementing a stack which supports IPv6 enabled RPL [69] for CSN communications is potentially feasible. However, in broad terms, the power use, additional overheads and low mobility basis of 802.11 based protocols make such protocols generally unfavourable for CSNs. Mohammad et al. point to C2CNet (Car-to-Car Net) as the emerging state-of-the-art standard within the field VANETs. C2CNet's protocol stack makes heavy use of 802.11 standards.

The field of FANETs (Flying Ad-Hoc Networks), a sub-domain of VANETs, is more relevant to CSNs. A survey by Bekmezci et al. [92] introduces FANETs in the context of both MANETs and the VANETs. The authors deal primarily with unmanned aerial vehicles (UAVs). Capability and resource constraints are more central within the field of FANETs. This is particularly true where long haul UAVs and small scale drone based applications are concerned. Bekmezci et al. highlight the departure from the use of standard 802.11 MAC layer protocols for FANET applications as the field has developed. The authors point to the development of tailored FANET MAC protocols, many of which aim to take advantage of advancements in directional antenna design. At the network layer, both DSR and a modified AODV [93] are noted along with other well-known MANET routing protocols such as GPSR [94] and OLSR [95]. Due to the use of time slots for route discovery, this modified AODV protocol has similar properties to the combined behaviour of this work's proposed protocols.

The work of Bekmezci et al.'s provides an overview of FANETs which details the use of customized MAC protocols and mainstream MANET protocols. One FANET related project cited by Bekmezci et al. called the Cooperative Autonomous Reconfigurable UAV Swarm (CARUS) project [96] is worth mentioning briefly. The project focuses more on application layer coordination and formation flying. However, due to several notable parallels with CSNs the project merits investigation for future developments on CSN formation flying and cooperative observation.

Bekmezci et al. place value on cross-layer architectures (CLO) within in the field of FANETs. CLO, as discussed, is an important topic within WSNs and its applicability within the field of FANETs further reinforces its importance to the future of CSNs. Bekmezci et al. note works that take advance of cross-layer clustering and scheduling through the cross-layer sharing of attitude and antennae related information in order to improve performance [97, 98].

The state of art in MANETs provides an insight into several potential routing protocols which may be employed in addressing the CSN PvTP trade-off. The FANET sub-domain is found to have the highest relevancy to this work. Examining MANET prior art augments and reinforces findings relating to WSNs. Such prior art provides important context to the existing state-of-the-art in CubeSat communications and this work's proposed protocols.

2.3 CubeSat Communications

Prior to the development and flight of the first CSN related mission several published works examined the inter-communication of CubeSats and CSNs. Challa and McNair of the University of Florida provide extensive explorations of distributed applications implemented upon CSNs [99-102]. These works are out of the scope of this project due to their focus on CSNs applications rather than CSN communication protocols.

The most relevant published work in the area of CubeSat communications is a survey by Radhakrishnan et al. [10]. This survey provides this work's primary source for the exploration of state-of-the-art CubeSat communications. Radhakrishnan et al. detail several relevant works relating to CSN MAC and routing protocols.

Radhakrishnan et al. provide an overview of some of the common terms used when referring to physical CSN formations. A 'trailing' formation, sometimes referred to as 'leader-follower', involves a single orbit chain of craft. A "cluster" of satellites generally implies a collection of satellites in multiple orbits which maintain some fixed formation. A "constellation" formation focuses on coverage of the earth's surface. Communications and GNSS constellations typically seek to achieve complete coverage or 'visibility' of key terrestrial regions. The term 'swarm', is also often misused. A swarm is not a satellite formation in the same sense as a cluster or constellation. Quoting Sundaramoorthy et al. "a satellite swarm is a group of identical, minimal, self-organised (self-functioning) satellites in space that achieve a common objective with their collective behaviour" [103]. Radhakrishnan et al. adopt the same definition as Sundaramoorthy et al. A similar concept to that of a swarm is the concept of 'fractionated' satellites wherein "the functionalities of a single large satellite are distributed across multiple modules, which interact using wireless links" [103] (Figure 15). This work's hypothetical mission involves a CSN in a cluster formation operating as a swarm.

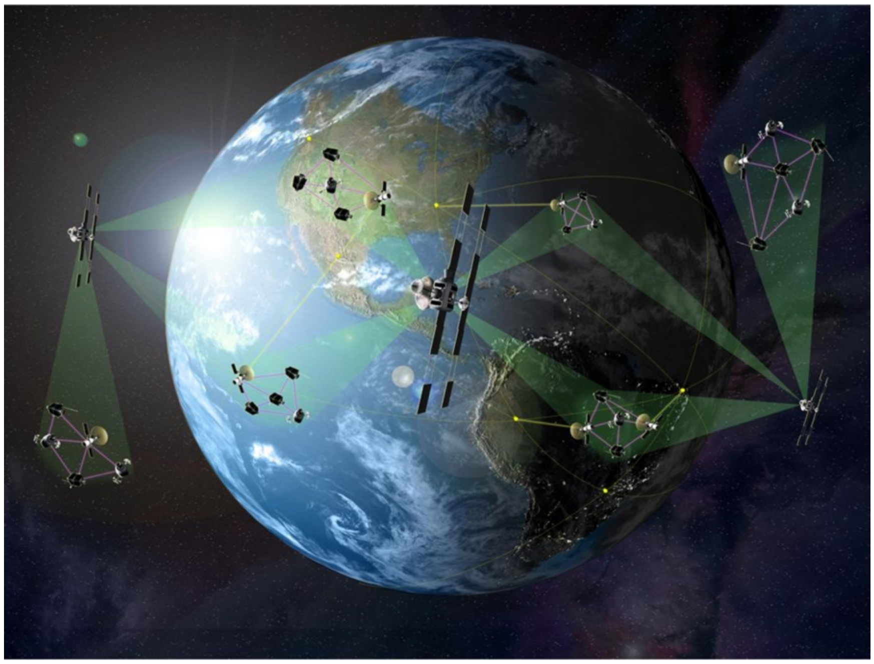


Figure 15. A rendering of the F6 DARPA fractionated satellite concept. Mission payloads exists independently of other core systems such as S2G communications. Image Credit: DARPA

2.3.1 Physical Layer

Although aspects of the physical layer of the OSI reference model are not of primary concern in this work it is nonetheless worth noting some of the findings of Radhakrishnan et al. When referring to maximizing data rates, Radhakrishnan et al. recommend a focus on increasing bandwidth rather than reducing the signal to noise (S2N) ratio. They also cite that higher S2G data rates can be achieved by transmitting in bursts rather than continually [104]. In terms of modulation and coding schemes employed at the physical layer, Radhakrishnan et al. cite Binary Phase Shift Keying (BPSK) as the current state-of-the-art for small satellites. Quadrature Phase

Shift Keying (QPSK) and offset-QPSK are noted for potential future development provided additional bandwidth balances out increased power requirements.

A considerable number of CubeSat communication related works focus on antenna design. Radhakrishnan et al. point to Gamalink [42] for its use of an advanced antenna. No further information regarding Gamalink is presented. Single patch S-band (2 – 4 GHz) antennae are highlighted as the current state-of-the-art. Also, the authors state that "a maximum (communication) distance of 1000 km between satellites can be achieved using a 3 W transmit power" using UHF (300 MHz and 3 GHz) radios [105]. This assertion seems dubious considering works previously discussed in relation to CubeSat communication capabilities. Radhakrishnan et al. express doubts regarding the suitability of complex MIMO and multi-patch antennae, instead recommending the use of multiple simple patch antennae. Radhakrishnan et al. state that links between satellites are generally full duplex and favour Time Division Duplex (TDD) over Frequency Division Duplex (FDD).

2.3.2 Data Link Layer

The data link, or simply 'link', layer has been discussed primarily in relation to Medium Access Control (MAC). Although the focus of this work continues to be placed on medium access control, it is worth noting other duties of link layer entities which include: packet framing, synchronization, error control, flow control and MAC addressing.

MAC protocols have a considerable effect on energy efficiency, network scalability, channel utilization, latency and throughput. There are two main classifications of MAC protocol: contention based and contention free. Contention based protocols, such as Carrier Sense Multiple Access (CSMA) derivatives, rely on detecting when the medium is in use and when two signals have caused a collision on the medium. Contention free protocols seek to completely avoid the need to detect medium use or collisions. Such protocols generally achieve contention freedom by allowing multiple agents to communicate at once without by logically dividing the medium such as with Code Division Multiple Access (CDMA) or Frequency Division Multiple Access (FDMA) (Figure 16).

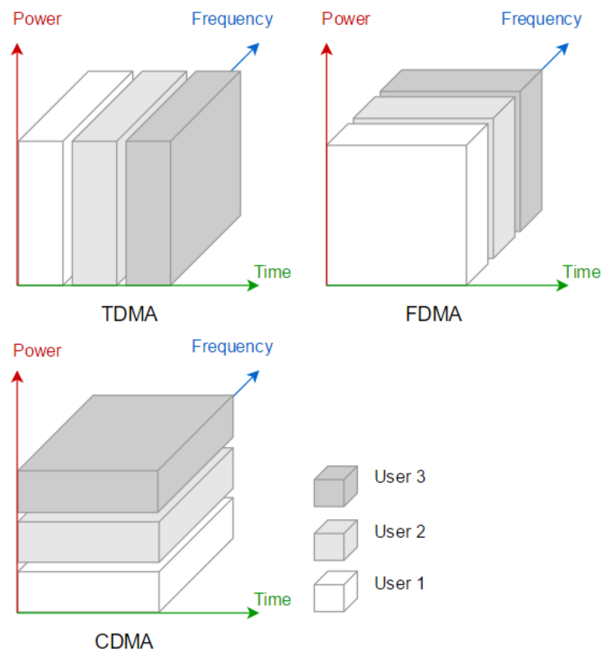


Figure 16. A comparison of common contention free MAC schemes. In CDMA, a 'chip', or code, is used to ensure that signals on the medium are orthogonal and therefore cannot collide.

Several MAC protocols are discussed by Radhakrishnan et al. in relation to small satellite communication. One approach attempts to adjust IEEE 802.11 physical and MAC standards for communications in LEO [106]. As discussed in relation to VANETs, several aspects of 802.11 based standards are unsuitable for space-bound communications. The modified 802.11 based approach addresses issues relating to inter-frame spacing (IFS). In LEO networks propagation delays may be in the order of milliseconds and can often be difficult to predict prior to communication. Using known propagation models and GNSS based information the proposed modified 802.11 approach adjusts IFSs and contention windows to fit communication delays. The distributed adjustment of IFSs and contention window may introduce considerable complexity, especially in large networks. The designers of this modified 802.11 approach assert its feasibility for use in LEO communications. However, the modifications provide a workaround to a problem left-over from terrestrial communications that need not exist in the first place.

A work, led by Radhakrishnan, explores the use of a CSMA style MAC protocol [107]. A protocol using CSMA with Collision Avoidance (CSMA/CA) is examined. The proposed protocol makes use of control packets to address problems such as the hidden node problem. In the hidden node problem, some node *B* can hear nodes *A* and *C*. However, nodes *A* and *C* cannot hear one another. Both *A* and *C* may sense the medium and incorrectly determine it as being free and attempt to communicate to *B*. Request To Send (RTS) and Clear To Send (CTS) packets control packets can ameliorate this issue. A and C can avoid communication with *B* until receipt of an appropriate CTS packet from *B.* This approach ensures that the medium is free from the perspective of the receiving node. In their assessment of this CSMA/CA approach, Radhakrishnan et al. conclude that the protocol is best applied in situations with low frequency communications and tightly grouped formations.

Researchers at the University of Delft propose a CDMA based MAC protocol for use in "Precision Formation Flying" (PFF) missions [108]. The proposed protocol employs a form of half-duplex CDMA which allows networks to adaptively scale and reconfigure as new members are introduced. The CDMA scheme was shown to have adverse effects on the ranging and navigation functions required for PFF missions. As such, the protocol's designers recommend the use of adaptive transmission power control mechanisms. PFF missions require high frequency low latency communications. Given this, CDMA may be the best approach. However, for the CSN PvTP trade-off, the additional power requirements of pure CDMA are not matched by obvious benefits in throughput.

Radhakrishnan et al. discuss a protocol proposed by Chen et al. called "Load Division Multiple Access" (LDMA) [109]. LDMA is a hybrid MAC protocol which utilizes a mix of Time Division Multiple Access (TDMA) (Figure 16) and CSMA in an attempt maximize channel (medium) utilization. LDMA allows network elements to operate in two different modes: High Contention Level (HCL) and Low Contention Level (LCL). HCL mode is used in response to high levels of communication which may result in numerous collisions and vice versa for LCL. In HCL mode TDMA is used. TDMA protocols rely on a schedule of time slots shared among network members (Figure 16). Each time slot is typically assigned a single owner. During a time slot, only the slot owner may transmit data. There are many different flavours of TDMA. For instance, schedules and slot ownership may be fixed or may be negotiated between nodes in a distributed manner. LDMA uses a fixed TDMA scheme.

In LCL mode, a version of CSMA is used. Several nodes in LDMA may be in differing modes at any one time. As such, nodes in LCL mode gives priority to the owner of the current time slot whenever collisions are detected on the medium. In response to collisions nodes will generate "conflict frames". Nodes switch between LCL to HCL based on the number of conflict frames detected.

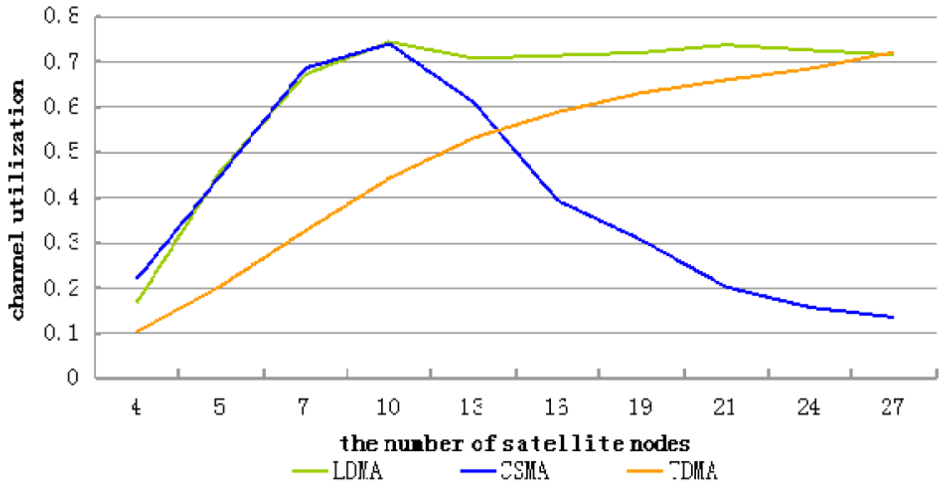


Figure 17. Cannel utilization (Vertical axis), measured from 1.0 (100%) to 0, compared to network size (Horizontal axis) for LDMA, pure CSMA and pure TDMA. Image Credit: [109]

Channel utilization generally refers to the percentage of time for which the common radio medium is used for communication of data. The communication of data in this case being distinct from the communication of protocol control information. Cannel utilization is best measured under steady state “saturated” conditions wherein a node always has a data packet queued to send. When correctly measured, cannel utilization is a key indicator of the overall throughput performance of a network. Chen et al. compare LDMA to pure TDMA and CSMA protocols through simulation. A graph representing their findings is shown in Figure 17.

LDMA presents itself as a strong candidate for use in attempting to optimize throughput for CSNs. As communication frequency drops the protocol mirrors a pure CSMA approach. Similarly the protocol mirrors a pure TDMA approach as activity increases (Figure 17). The energy consumption profile of the protocol merits further investigation. The protocol requires nodes to be promiscuous in order to overhear conflict frames and mode change broadcasts. Leaving radios constantly in receiver mode will incur an energy consumption penalty over time. In comparison, certain TDMA protocols allow nodes to completely sleep their radios for periods under certain conditions.

Two further hybrid approaches are discussed by Radhakrishnan et al.: An FDMA/TDMA (F/TDMA) hybrid and a CDMA/TDMA (C/TDMA) hybrid. F/TDMA is based on WiMedia [110]. The protocol introduces the distributed management of heterogeneous network state as well as two-dimensional "super frames" in place of TDMA time slots. C/TDMA employs a cluster based approach which requires cluster "slaves" to use CDMA and cluster "masters" to use TDMA [111]. Comparatively the C/TDMA protocol provides many of the same properties as F/TDMA protocol without F/TDMA's potentially prohibitive levels of complexity.

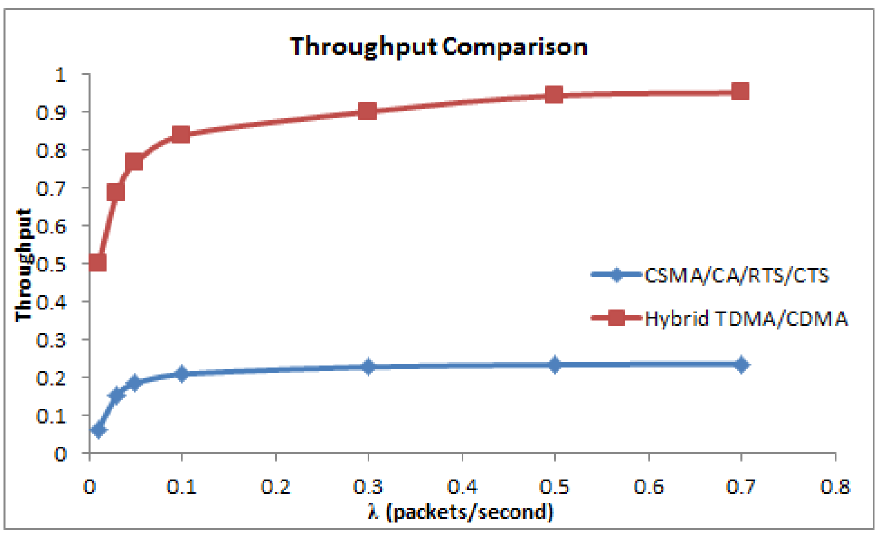


Figure 18. Throughput here is measured as the amount of time spent transmitting data divided by the amount of time available to transmit data. Image Credit: [111]

The C/TDMA protocol is chosen as the starting point for this work's proposed MAC protocol. The details of the protocol will be discussed at length in chapter 3. The protocol was chosen as it makes explicit allowances for energy awareness and the improvement of throughput. For a given simulation scenario involving a trailing formation, C/TDMA out-performs Radhakrishnan et al.'s

CSMA/CA approach (Figure 18). The protocol's formation of clusters is based on the energy available to nodes. Nodes with higher levels of remaining power are elected as cluster "Masters" which act as routers between clusters. This allows non-master nodes to conserve energy.

2.3.3 Network Layer

The primary responsibility of the network layer is routing, although packet forwarding and address handling are also important network layer activities. Routing protocols affect the discovery and selection of optimal routes within a network. Several generally applicable approaches to the discovery of routes are discussed in the context of MANETs. Unlike discovery, the selection of optimal routes is often highly application dependent. For instance, in the context of this work, there may be benefits in avoiding routes which rely on CubeSats with low amounts of remaining power. The approaches to determining and maintaining optimal routes differentiate many of the routing protocols proposed for small satellites. There is relatively less published work relating to the routing protocols than to MAC protocols. As evident from the Nodes and Tianwang-1 missions, the current state-of-the-art for CSNs are small one-hop networks. Such networks don't need to perform route discovery or chose between multiple routes. As such, the network's performance is predominantly determined by layers below the network layer, such as the link layer.

Radhakrishnan et al. discuss a number of routing protocols in their survey of inter-satellite communication for small satellites [10]. The authors highlight routing approaches which have been adapted in the past for use with larger satellites such as the Border Gateway Protocol (BGP) [112] and Multi-Layered Satellite Routing (MLSR) [113]. The authors make no comment on the suitability of these approaches for small satellites. This is a theme throughout the authors' discussion of routing protocols which reflects the scarcity of relevant research in this area. For instance, the authors discuss delay tolerant networking (DTN) at length. DTN has been proposed recently by some for small satellite applications. However, notable works on DTN for CubeSats have yet to be published.

An approach proposed by Bergamo et al. involves each node classifying its neighbours as either 'new' or 're-occurring' [114]. The approach is similar to that of AODV however the classification of neighbouring nodes may reduce the overall frequency of route discovery. The protocol mainly focuses on the synchronization of larger satellites which maintain regular controlled orbits. Nonetheless, this protocol may have benefits in situations involving CSNs composed of multiple swarms in disparate orbits.

Of the routing protocols discussed by Radhakrishnan et al. few are obviously suitable to CSNs and none deal directly with the balance of energy efficiency and throughput. As such, prior art relating to MANETs and WSNs informs the choice and adaptation of this work's proposed routing protocol.

Radhakrishnan et al. recommend further work on cross-layer optimization (CLO) and the introduction of protocols which adapt naturally to predictable topology changes. No recommendation is provided for further work on routing. Radhakrishnan et al.'s survey is a valuable resource when approaching the domain of CubeSat communications. The authors cover the physical and data link layers well but fail to discuss and identify the gaps in relation to the network layer. Also, the authors provide considerably more content for works on which one or more of the survey's authors were involved which may indicate a degree of bias.

2.3.4 Other Works

Wong et al., operating mainly out of NASA's Goddard Flight Centre, examine a potential future for CSNs. Wong et al. propose that S2G communication be performed indirectly through relay with existing space bound communication networks [11]. This concept is further explored for deep space missions in the preliminary development of ESA's COPINS mission [60].

Another survey style paper on inter-satellite link for CubeSats by Budianu et al. [9] published in 2013 provides an overview of the field. Budianu et al. pay more attention to antenna design and link budget analysis than to areas relevant to this work.

2.4 Other Areas of Note

Alongside the primary areas of concern there are secondary areas which are deemed to be less relevant to this work. These secondary areas further illustrate the considerable context which must be considered when exploring aspects of CSNs.

2.4.1 Energy Aware Scheduling

Energy aware scheduling is an active area of research in terrestrial research domains, especially in relation to WSNs [115]. Despite the growing popularity of small form factor satellite missions, there are few notable energy aware scheduling related publications in the domain. However, new insights into the area were produced in 2016 as a result of the GomX-3 mission [116]. GomX-3 was designed by a private Danish company, GOMSpace and flown by ESA. Following the mission's success, mission designer published a work entitled "Battery-Aware Scheduling in Low Earth Orbit The GomX-3 Case" [61]. The work outlines an approach which adaptively models and predicts power usage to produce an activity schedule. This schedule is intended to optimize overall craft energy consumption. In the context of CSNs, the sharing of individual CubeSats schedules could lead to the generation a CSN activity schedule which optimizes S2S and S2G communication energy efficiency.

2.4.2 Delay Tolerant Networking

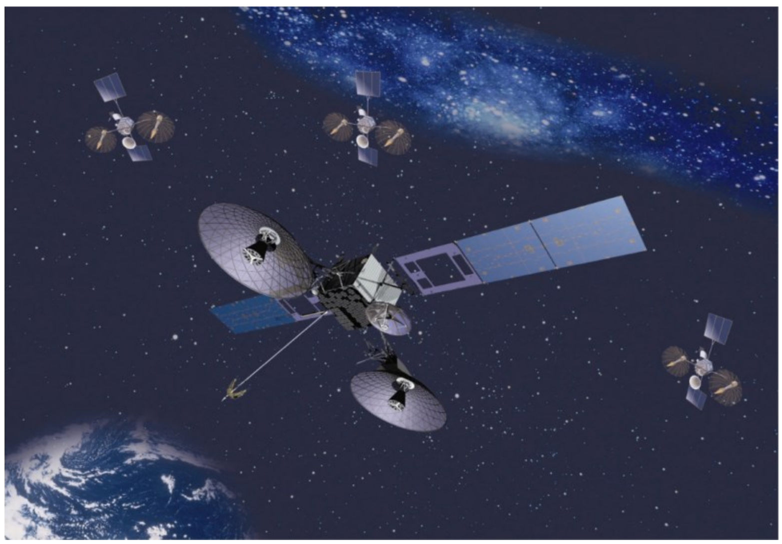


Figure 19. NASA Tracking and Data Relay Satellites (TDRS) which form the backbone of NASA's deep space network. Image Credit: NASA

Delay tolerant networking (DTN) approaches have been employed successfully over the past decades to solve many of challenges of inter-planetary communication [117]. NASA's deep space network is a notable DTN success case [118] (Figure 19). Although CSN's don't face the same magnitude of challenges presented by inter-planetary communication many, including Radhakrishnan et al., point to DTN as important to the future of CSNs. DTN has the distinct advantage of being tested with larger satellites networks and developed by experts within the space industry.

Chapter 3: Proposed Protocols

Two link layer MAC protocols are identified as potential candidates for the basis of this work's proposed MAC protocol: LDMA [109] and C/TDMA [111]. Of these, C/TDMA was chosen. However, both protocols merit investigation for application within CSNs. LDMA offers the potential for the best aspects of both CSMA and TDMA. Whereas, C/TDMA has the ability to selectively reduce the energy consumption of certain nodes through cluster formation. The choice of C/TDMA was motivated primarily by the protocol's relative simplicity and a greater availability of implementation information.

This work makes several changes to C/TDMA as specified by Radhakrishnan et al. These changes are primarily intended to enable nodes to opportunistically conserve energy. The final protocol is referred to as "CubeMac" for convenience. CubeMac's operation remains founded in C/TDMA with many of the changes made drawing inspiration from EDSN's Cpt/Lt protocol.

The development of Gamalink [42] and the recommendations of several domain experts [10, 13, 58] indicate that multi-hop networks are the next stage of development for CSNs. For such future networks, this work proposes a protocol based on the DYMO [83] routing protocol. This choice was primarily motivated by the availability of an existing implementation of DYMO for OMNeT++. Without an existing implementation, it is unlikely that a suitable routing protocol could have been fully addressed given available development resources. Modifications made to this DYMO implementation focus mainly on resolving pre-existing implementation issues. This work also introduces the concept of a "Ground Master" (GM). A GM is similar in function to an

EDSN Captain, in that, only a GM may conduct S2G communication. This work's implementation of DYMO is referred to as D^3 (DYMO Cubed).

This work places the majority of its focus on CubeMac. Compared to CubeMac, the customization and the choice of basis for D^3 are less informed by the current state-of-the-art. This is partially a result of the scarcity of prior art relating to CSN routing protocols. Nonetheless, DYMO represents a suitable candidate for the basis of this work's proposed routing protocol. As discussed in section 2.2.2, DYMO is a reactive MANET routing protocol based on AODV. DYMO's reactive nature allows for the examination of the effects of intermittent ground access through the on-demand discovery of S2G links. Without a reactive protocol such as DYMO this aspect of CSN's would have required an idealized simulation approach reducing the fidelity of this work's results. Both D^3 and CubeMac combine through elements of cross-layer optimization to form this work's primary contribution.

3.1 CubeMac

The CubeMac protocol builds upon the work of Radhakrishnan et al.'s C/TDMA protocol. CubeMac is designed with direct consideration to the PvTP trade-off and the chosen hypothetical mission. CubeMac focuses on data collection. In an attempt to reduce energy consumption CubeMac avoids distributed decision making. CubeMac propose several potentially beneficial additions to the work of Radhakrishnan. The following sections detail the operation of CubeMac and discuss its potential strengths and weaknesses.

3.1.1 TDMA

TDMA, as introduced in section 2.2.2, can be implemented with varying degrees of adaptivity. TDMA time slot owners can be dynamically assigned through distributed negotiation [119]. Approaches also exist which allow nodes to opportunistically share time slots and dynamically adjust the length of time slots [120]. Radhakrishnan et al. take a purely static approach to the TDMA aspects of C/TDMA. In this static approach time slots cannot be shared and have fixed lengths and owners. Although the avoidance of more adaptive approaches is not discussed directly by C/TDMA's designers, static TDMA has several desirable properties.

The addition of adaptivity to TDMA protocols introduces the need for distributed decision making and consensus. These added requirements introduce varying degrees of overheads and delay. Propagation delays in LEO may be in the order of milliseconds. Also, CubeSat clocks may drift apart by up to 12ms [57]. Distributed decisions making and consensus approaches under such conditions, although not infeasible, may incur potentially unacceptable time and energy costs. The Cpt/Lt protocol, although adaptive in its assignment of Cpt and Lt roles, uses a static TDMA approach i.e. the schedule and duration of time slots and their respective owners is fixed and known to all network members.

3.1.2 Cluster Formation

Static TDMA is coupled with CDMA to form Radhakrishnan et al.'s C/TDMA. This integration is achieved by the introduction of clustering. Clustering is introduced in section 2.2 in the context of WSN-MEs. Using a "centrality algorithm", which is not specified further by Radhakrishnan et al., a CSN may divided into several clusters as in Figure 13. Within each cluster, a "master", similar in function to a cluster head, is elected. The remaining nodes in the cluster assume the role of “slaves”. Inter-cluster communication is handled solely by cluster masters (Figure 20). Master-to-master and master-to-slave communication takes place using TDMA. Slave-to-master communication uses CDMA. A special time slot is dedicated to the use of CDMA during which all slaves communicate with their respective cluster masters (Figure 21).

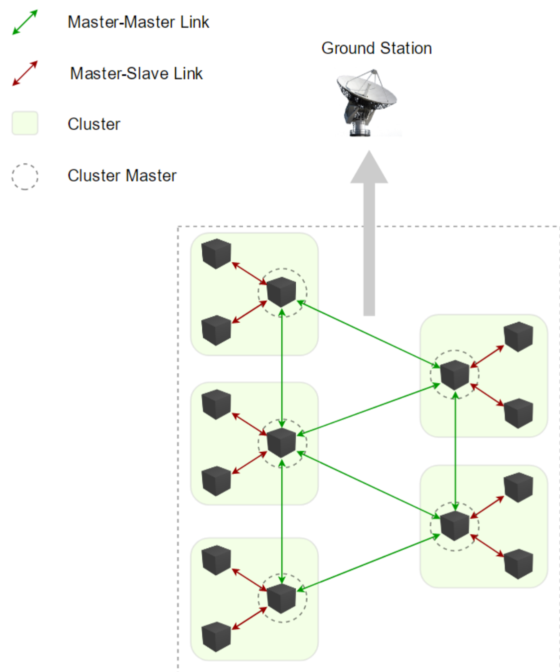


Figure 20. Inter and intra cluster communication in CubeMac. Note that slaves only communicate with one another via cluster masters.

Radhakrishnan et al. specify two distinct types of TDMA frames. A frame in TDMA nomenclature is a group of time slots which follows some fixed repeatable pattern (Figure 21). In the specification of C/TDMA one frame is dedicated to ‘uplink’ and the other to ‘downlink’. The use of the term ‘downlink’ in this case should not be confused with S2G communication which is often referred to as ‘downlinking’. Radhakrishnan et al. do not include a ground station in their simulation and assessment of C/TDMA. During the uplink frame, slots are dedicated to allow slaves transmit to their cluster master and for cluster masters to transmit to one another. During the downlink frame, cluster masters focus on transmitting to slaves within their clusters.

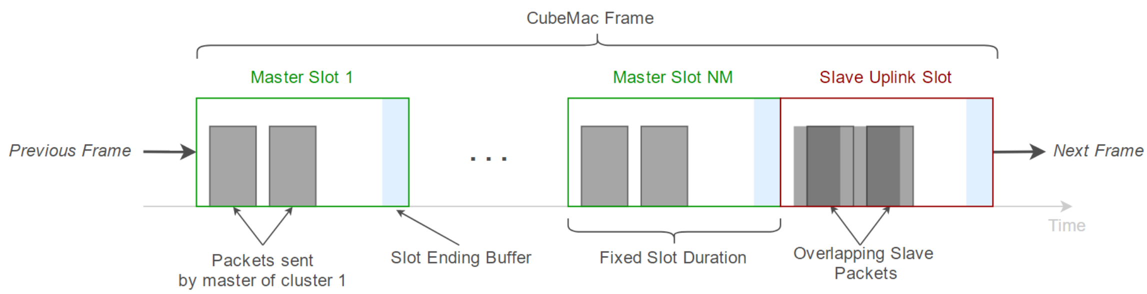


Figure 21. One complete CubeMac frame. "Master Slots" are collision free through TDMA. Communication during "Slave Uplink Slots" is made collision free using CDMA. Frames repeat indefinitely.

As discussed in section 2.2.1 this work is concerned with data collection and not data dissemination. As such, CubeMac implements a modified version of the Radhakrishnan et al.'s uplink frame (Figure 22). In this work's simulation scenarios, only CubeMac masters may be selected for S2G communication. A selected master is referred to as a "ground master" (GM). Slave packets are routed first to their cluster master. If this master is not a GM, these packets will be forwarded through neighbouring masters until reaching a GM (Figure 22). CubeMac's operation is unaffected by D^3's behaviour. D^3 however is affected by the assignment of CubeMac's master and slave roles.

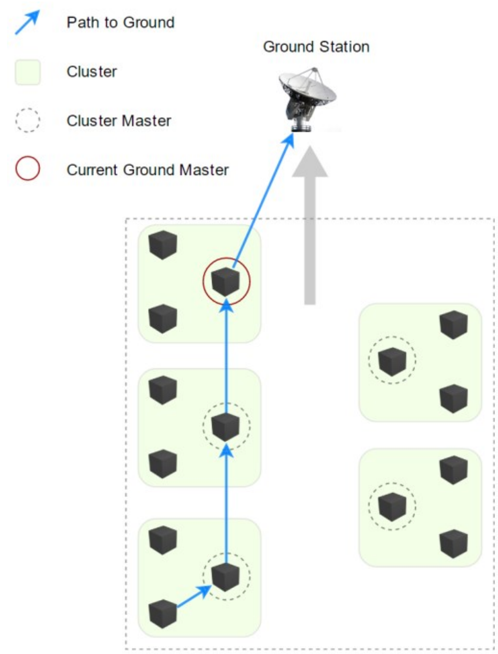


Figure 22. All routes from slaves to ground require one or more masters. The Ground Master always constitutes the final hop on a path to ground.

3.1.3 CDMA

Simulations of CubeMac employ a simplified implementation of CDMA. In simulations, only the additional energy costs and parallel communication of CDMA are modelled. This approach is deemed sufficient for the purposes of the exploring CubeMac's effects on the PvTP trade-off.

CDMA based protocols combine a 'chip', or 'code', with messages in order to form a final signal which is spread across a greater bandwidth (Figure 23). Multiple coded signals may share a wireless medium without collision or interference, provided appropriate codes are used. There are numerous other benefits to CDMA approaches including enhanced resistance to certain types of noise, fading and jamming. By sharing codes between nodes, messages may be retrieved from spread signals using the code which initially spread the message.

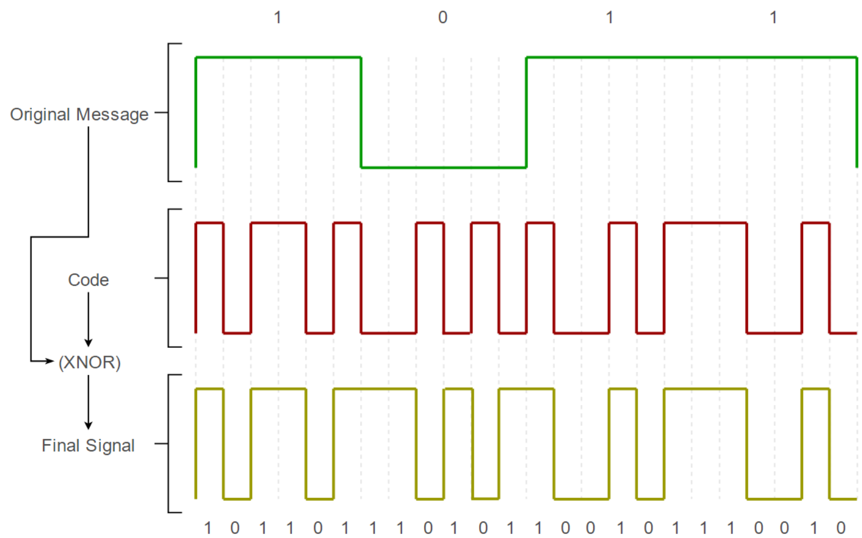


Figure 23. A CDMA approach to spreading an initial message over a greater using an example code. The original message contains four bits however the final signal is 24bits in length due to the 24bit code used.

Parallel communication through CDMA comes at the cost of added complexity and overheads. As shown in Figure 23, spread signals require greater amounts of bandwidth than the 'raw' message which they contain. There are also added computational overheads due to message encoding and decoding. These bandwidth and computational overheads result in an increase energy consumption over non-coding MAC schemes. CDMA also generally requires more complex and energy demanding modulation schemes such as those based on Quadrature Amplitude Modulation (QAM) [121].

In this work's simulation of CDMA, coding, modulation and spreading are ignored. Nodes simply receive messages unencoded and ignore any potential collisions. Simulated transmitters 'using' CDMA incur a higher energy cost. This simplification was motivated largely by the absence of an existing suitable CDMA implementation for OMNeT++.

3.1.4 Frame Structure

CubeMac's behaviour changes based on the role of a node. Masters uses TDMA for communication whereas slaves use CDMA. Assuming a network containing N_m clusters, and therefore N_m masters, each CubeMac frame will consist of $N_m + 1$ time slots. CubeMac assigns each cluster an ID from 1 to N_m. Each cluster master owns the time slot corresponding to its cluster ID. For example, the master of the cluster with the ID of 1 (M1) will occupy the time slot with the ID of 1. The final time slot (ID = $N_m + 1$), named the "uplink" slot, is reserved for slaves. During the uplink slot all slaves communicate with their respective cluster masters using CDMA. The completion of uplink slot marks the end of a CubeMac frame after which a new frame starts again at slot 1. During each frame nodes transition between three states: transmitting, listening and sleeping (Figure 24).

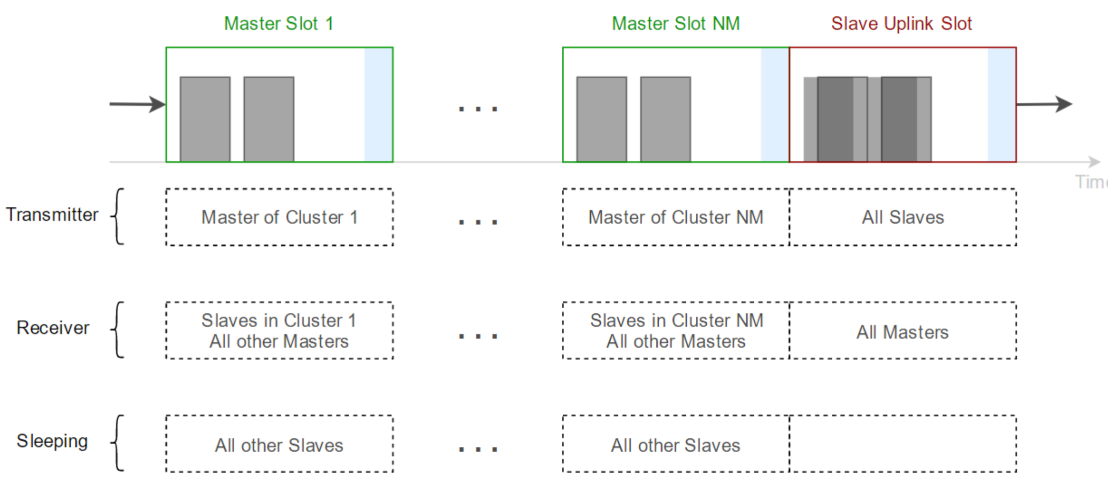


Figure 24 An illustration of the states which a given node assume during certain slots. No node may sleep during the slave uplink slot. These states are sufficient to allow multi-hop communication between all nodes within a network.

Cluster masters may use their slots to transmit to any node which is listening. This represents a departure from Radhakrishnan et al.'s C/TDMA specification. In C/TDMA, each master reserves a time slot for each neighbouring cluster master and each slave within its cluster. This change was necessary due to the omission of Radhakrishnan et al.'s 'downlink' frame. Without this change no transmissions could occur from masters to slaves. In this work, such transmissions are required by D^3 for the completion of route discovery.

Figure 21 and Figure 24 illustrate that each time slot ends with a short configurable buffer period (marked in blue). This buffer is included to combat clock drift between CubeSats. EDSN mission designers estimated the mission's maximum clock drift as 12ms. The buffer also acts to alleviate the effects of jitter. Masters and slaves must take this buffer period into account when predicting how many packets to attempt to transmit within any given slot.

3.1.5 Energy Saving Features

CubeMac introduces two additional features to C/TDMA both of which are intended to allow nodes to conserve energy. The first feature, which is illustrated in Figure 24, allows slaves to sleep during certain slots. Slaves only communicate with their cluster master. As such, slaves may sleep during any slot which is not the uplink or their cluster master's slot. Each slave maintains knowledge of the cluster within which it resides. As cluster IDs correspond directly the slots owned by the cluster masters, slaves may easily identify slots in which they may sleep.

The second energy saving feature allows nodes to sleep during time slots under various conditions. In implementing this feature CubeMac introduces a special "last" field within packet headers. The "last" field allows transmitting nodes to indicate that a packet is the last they intend to send during a slot (Figure 25). CubeMac also introduces a timeout period and "No Data" packets, which are not specified in C/TDMA (Figure 25).

CubeMac's optional "no data" (ND) packets and ND header field assist nodes in conserving energy during time slots (Figure 26). A node with no packets on its send queue, may broadcast an ND packet at the start of an owned slot. Such packets are intended to allow nodes 'waiting' in receiver mode to sleep prior to observing a packet timeout period (Figure 26). To reduce the overheads of transmitting an additional packet, ND packets are made as small as possible. Using ND packets, timeouts and the "last" packet field CubeMac presents several conditions under which a node may sleep during a time slot.

Masters may sleep when:

1. No packet is received from a slot owner or from any slave during an uplink slot for a period longer than the configured timeout period
2. During a non-uplink slot, the slot owner sends an ND packet or a packet marked as "last"
3. During an uplink, all cluster slaves transmit an ND packet or a packet marked as their last
4. An ND packet or a packet marked as "last" has been transmitted

Slaves may sleep when:

1. No packet is received from the cluster master during the cluster master's slot for a period longer than the configured timeout period
2. During the cluster master's slot, the cluster master transmits an ND packet or a packet marked as "last"
3. An ND packet or a packet marked as "last" has been transmitted

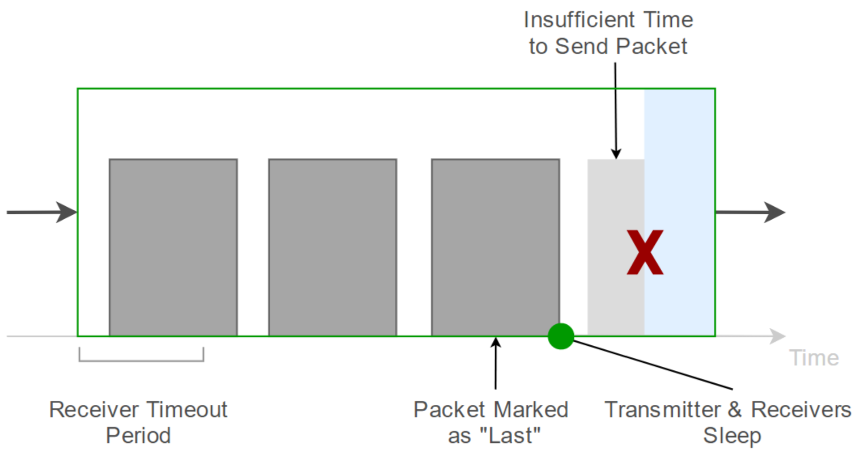


Figure 25. The last packet transmitted by any node within a time slot will contain a flag indicating that no further packets should be expected.

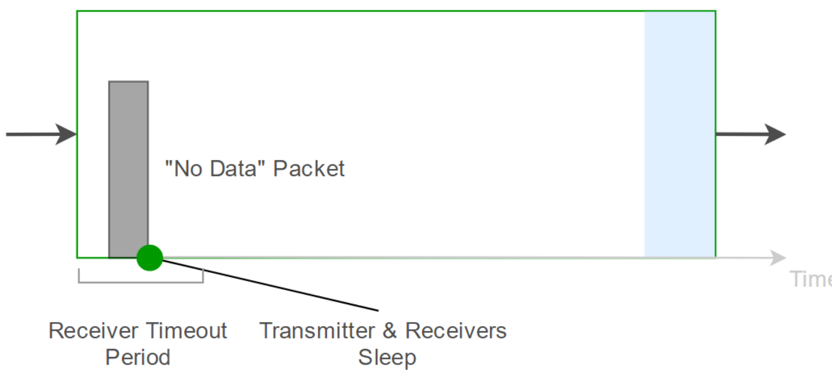


Figure 26. Nodes may generate a "No Data" packet when they have no data to send during their slot. Sending this packet incurs an energy penalty but this may be out-weighed by allowing multiple nodes to sleep prior to a timeout.

The only tangible overheads introduced by the added energy saving features are those of the added packet header fields. The "last" and ND fields require one additional bit and the cluster ID field requires $\log_2(N_m)$ bits. Crucially these energy saving features should not reduce S2G throughput. These features take advantage of the correspondence between slots and cluster IDs as well as the master-slave relationship without reducing S2S communication windows.

3.1.6 Drawbacks

The restriction of slave communications to cluster masters is fundamental to CubeMac. To lift this restriction would require introducing additional time slots for each slave within a cluster at which point CubeMac essentially approximates TDMA. Where data collection is concerned, allowing slaves to communicate with one another offers no obvious benefit as only masters may communicate to ground stations. It is assumed that cluster masters are elected using a similar approach to EDSN's captaincy election approach (Section 2.1.2)

If a new packet is generated immediately following the end of a node's time slot the packet must wait for an entire frame before being sent (Figure 27). Packets generated immediately after a node sends an ND or "last" packet will face even longer delays. However, the effects of ill-timed packet generation are of little concern in the context of this work. Ill-timed packet generation will affect packet latency but not S2G throughput. In this regard CubeMac sacrifices latency in favour of reduced energy consumption.

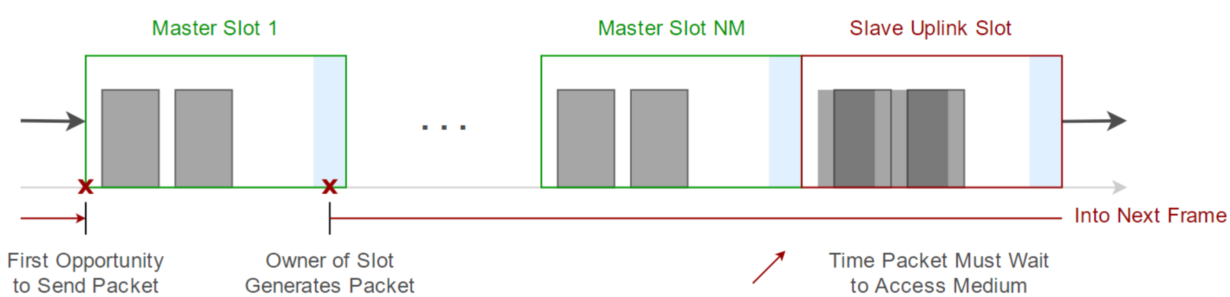


Figure 27. Packets generated within a buffer period or directly following the end of a slot must wait a minimum of approximately N_m slot durations before transmission.

Similar to the ill-timed packet generation issue is that of slot saturation. Consider a scenario where only one node, a master for instance, is generating packets. If this node consistently generates more packets than it can send during its time slot the medium will be poorly utilized. In CubeMac, this master cannot borrow time from another time slot to transmit its additional packets even though all other time slots may be effectively empty. LDMA overcomes this issue through the use of CSMA in such low contention scenarios. However, yet again, the impact of this issue is reduced as a result of the context of this work. The chosen hypothetical mission assumes that all nodes carry scientific instruments and generate packets containing science data at the same rate. This creates a consistent loading condition across all nodes.

The use of global time slots also creates an issue in relation to CubeMac. Nodes within CSNs are likely to be widely dispersed. Given all nodes are within range of one another, TDMA will succeed in avoiding all potential conflicts without any 'waste'. However, in spatially disparate networks, such as CSNs, waste will occur when nodes which cannot interfere with one another do not share time slots. In such cases, nodes can become 'isolated' from the current time slot and cannot perform any useful communication. This is a direct result of the global assignment of time slots. In cases where time slots are dynamically allocated at a local scale this isolation can be avoided. Once again this issue is addressed by the low contention level mode of LDMA. A protocol referred to as Lightweight MAC (LMAC) uses a dedicated slot assignment phase in order to assign time slots at a local rather than a global scope [119]. LMAC uses basic carrier sense based techniques during the assignment stage to overcome issues inherent with local time slot assignment such as the hidden node problem discussed in section 2.3.2.

The general nature of CubeMac makes it ill-suited for handling node failures. For instance, if a master's science instrument fails then an entire time slot will remain unused regardless of the saturation of other slots. If a master as a whole dies an entire cluster will become isolated from the network and ground (Figure 20). CubeMac has no facilities for acknowledging the receipt of packets and as such slaves cannot recognize when their master has died or drifted out of range. The lack of acknowledgement scheme is not fundamental to CubeMac and one could be added. Acknowledgments are omitted as this work concentrates on the PvTP trade-off without consideration of node failures.

Dynamic clustering and master election are important in addressing challenges related to node mobility and failure. Neither feature is implemented in this work's simulation scenarios. This work considers the PvTP trade-off during "steady-state" network operation. It is worth noting that both dynamic clustering and master election could be added to CubeMac as required. As masters have considerably more work to perform than slaves, periodic master selection and re-clustering based on available node battery capacities constitute crucial features of any real-world implementation of CubeMac. As discussed, EDSN's captaincy election mirrors that of cluster master election. Without regular master election, master nodes would quickly exhaust their available energy supplies and all slaves would become isolated from ground. Radhakrishnan et al. mention master election and clustering briefly but provide no implementation details of either.

3.2 D^3

D^3's operation is based on the DYMO routing protocol. As discussed in section 2.2.2, the design of DYMO is based on AODV. Like AODV, DYMO makes use of three different "route messages": route requests (RREQs), route responses (RREPs), and route errors (RERRs). DYMO's general use of these messages is the same as that of AODV which is detailed in section 2.2.2 and Figure 14. The remaining relevant aspects of DYMO's operation are discussed in the following sections.

3.2.1 Intermediate RREPs

In section 2.2.2, the concept of intermediate RREPs is briefly introduced in the context of AODV. DYMO employs the same approach as AODV in implementing this feature. A node receiving an RREQ for a given target may check whether it has a valid route for the target. If the node has a valid route it may generate an RREP to be sent to the originator (Figure 28). This RREP will be identical to an RREP as generated by a target receiving the same RREQ.

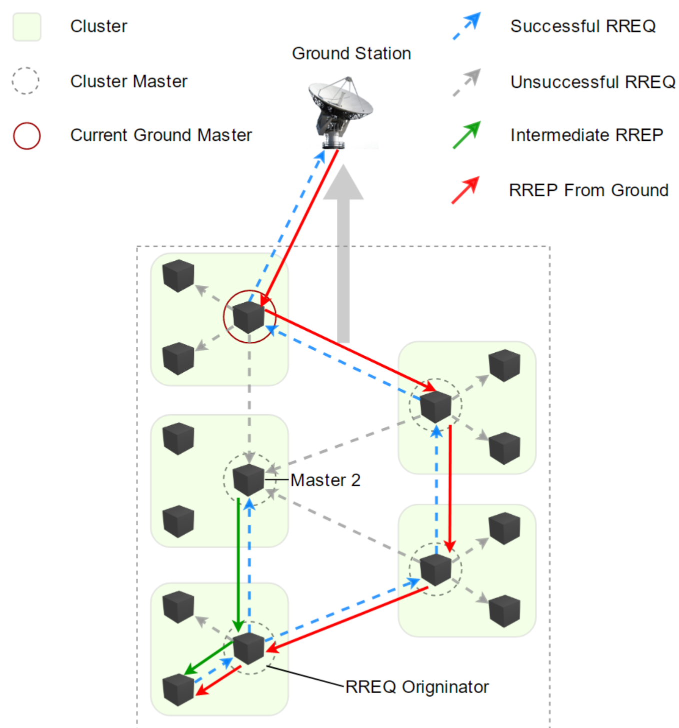


Figure 28. Master 2 generates an intermediate RREP as it already has a route to ground. The original RREQ for this intermediate RREP had to travel two hops as opposed to the minimal four hops to ground. The RREP from ground may replace the route generated from the intermediate RREP if its route has a lower cost.

Allowing intermediate RREPs can significantly reduce the amount of traffic generated during each route discovery attempt. Intermediate RREPs are especially impactful in the case of

CubeMac. All CubeMac slaves only communicate with cluster masters, as such the next hop for any route held by a slave will always be its cluster master. If the cluster master has a route for a target it can generate an RREP and reduce the need for slave RREQs to traverse the network.

In simulation scenarios, as this work is concerned with S2G throughput, all packets are routed to the address assigned to the ground station. In CubeMac, master time slots occur before the slave uplink slot. This arrangement increases the likelihood that a master will be able to generate an intermediate RREP and reduce RREQ traffic. However, there is one notable drawback to consider with intermediate RREPs. Reducing the propagation of RREQs to target nodes increases the likelihood that suboptimal or stale routes, held by intermediate nodes, are used.

3.2.2 RERRs

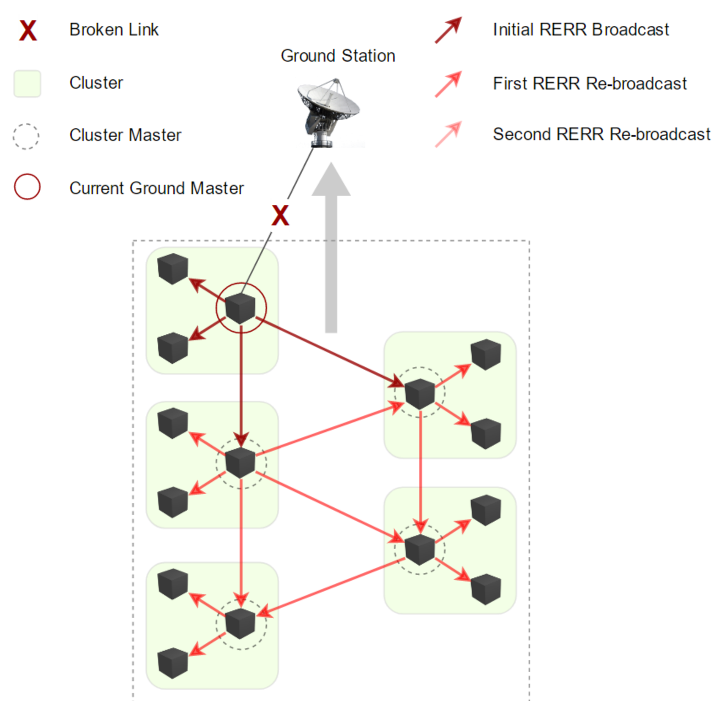


Figure 29. The broadcast based propagation of an RERR message throughout the hypothetical mission's CSN.

RERRs are generated in response to link breakages. Detecting such breakages often relies on packet acknowledgement schemes. As discussed, CubeMac does not utilize acknowledgement schemes and node failure is not within the scope of this work. Also, the specification of this work's hypothetical mission in section 1.2 states that nodes are assumed not to move relative to one another.

In this work, a node's motion relative to ground is the only source of link breakages. As will be discussion further in section 4.1.2, master nodes use a separate network interface for S2G communication. The S2G interface uses a MAC protocol based on CSMA rather than CubeMac. This CSMA protocol utilizes acknowledgements. As ground masters move out of range of ground this CSMA protocol signals DYMO that the S2G link has broken. This is the only mechanism by which links break within this work's simulation scenarios. In response to a link break signal DYMO will generate a RERR message. This message is broadcast throughout the network allowing all nodes to discard any routes to ground which they maintain Figure 29.

3.2.3 Sequence Numbers

Like AODV, DYMO utilizes monotonically increasing sequence numbers (SNs) to determine the freshness of a route. DYMO records the SN associated with each incoming route. Prior to replacing an existing route DYMO checks a candidate route's SN against the SN of the existing route. If the candidate's SN is lower than the existing SN then the candidate route is considered stale. By default, stale routes are discarded regardless of whether the route has a lower cost that the existing route. As a result of the relative motion of nodes such stale routes have a higher likelihood of being broken. It follows that routes with higher SNs have been established more recently. Such recent routes are less likely to have been affected by topology changes. SNs can also be used to detect routing loops. However, this is not the approach taken by DYMO.

3.2.4 Route Costs

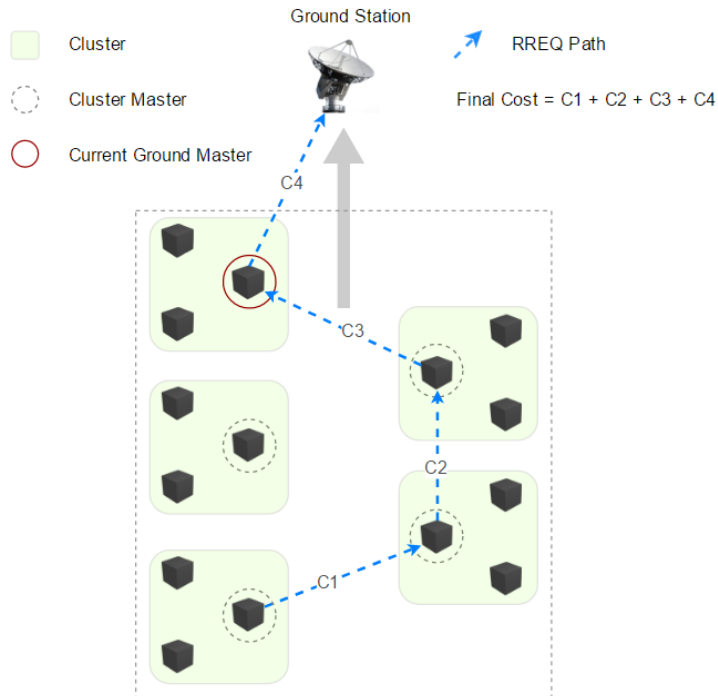


Figure 30. As an RREQ, or RREP, packet moves through a network it accumulates link costs within its 'metric' field.

DYMO allows nodes to record each route's 'cost'. These costs are central to DYMO's approach to routing loop detection. Route costs are established within route messages. For instance, as an RREP is transmitted from the target node to the originator node, each intermediate node adds some greater-than-zero cost value to the RREP's 'metric' field (Figure 30). The final value in this field represents the cost of the route described within the RREP. The value added by an intermediate node represents the cost of the link between the intermediate node and the node previously visited by the RREP (Figure 30).

The most common and basic cost value utilized is a route's "hop count". When using a hop count cost function intermediate nodes add '1' to the current cost field within RREPs. Originators will then favour routes which represent the shortest path to a given target. As discussed in section 2.2.2, energy metrics are a common and impactful source of link cost values. An energy aware cost function may utilize information regarding a node's a remaining battery capacity. For example, a node may use the following logic to determine a link's cost: if current battery capacity less than *X*: add '10' to link cost – else: add '1' to link cost. Provided all node's use the same cost logic, nodes will favour routes which avoid nodes with battery capacities below the threshold of *X*. If all nodes for all possible routes have battery capacities above this threshold, then routes with the lowest hop counts will be used.

Costs may also be based on other metrics such as those relating to the signal strength of a received RREP, the number of packets a node has waiting for transmission, and the time remaining before a node will be in range of ground. These approaches represent the application of varying degrees of cross layer optimization. It is also possible for CubeSats to use the GomX-3 energy aware scheduling approach, discussed in section 2.4.1. Using the GomX-3 approach, CubeSats could forecast upcoming energy availability and demands in order to determine the potential energy cost of announcing a given 'link' cost.

DYMO is capable of detecting routing loops through the use of route costs and the opportunistic establishment of routes. Unlike some other routing protocols, DYMO route entries only require a target address and a next hop address. In Figure 31, Master 2 (M2) receives an RREQ from the RREQ originator (RO) via route 1. Using the received RREQ, M2 records a route entry for the RO. When M2 receives the RREQ taking route 2 it compares the its cost metric to its existing route entry for the RO. As M2's route entry for RO has a lower cost, it can safely drop this RREQ. Without this logic rebroadcasts of this RREQ could continuing moving in loops until generated RREQs reach the original RREQ's hop limit.

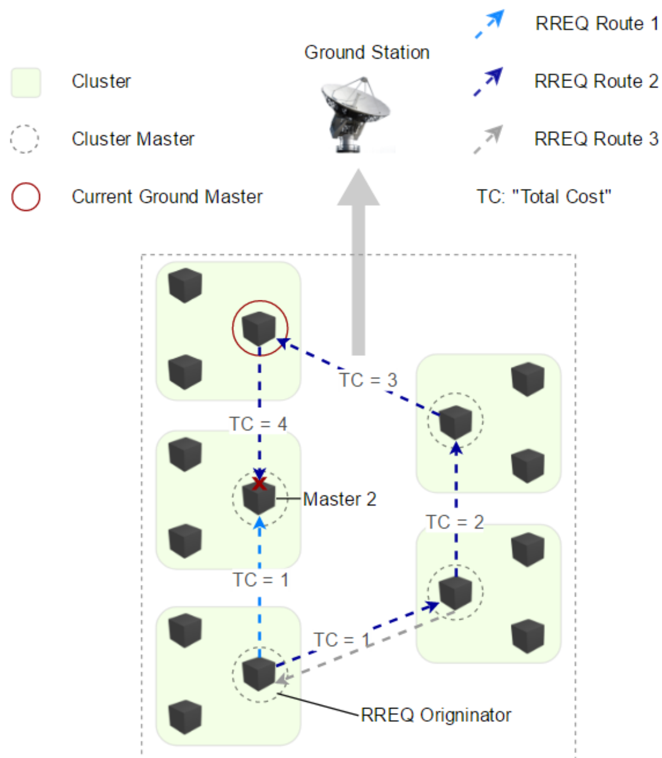


Figure 31. DYMO detects loops by comparing a route message's cost metric with relevant existing route entries.

DYMO implements a different approach to handle RREQ route 3. The RO broadcasts its RREQ which is received by the first hop on route 2. This receiver rebroadcasts the RREQ. This rebroadcast will be received by the RO. The RO drops this RREQ as the address in the RREQ's "originator" field matches its own address. Together, the approaches illustrated in Figure 31 allow DYMO to avoid routing loops of arbitrary lengths.

3.2.5 Discovery and Maintenance Patterns

DYMO's route messages, sequence numbers and route costs provide sufficient capabilities to discover, chose, and detect the breakage of routes. The remaining salient aspects of DYMO relate to the timing of route discovery and route maintenance. Like AODV, DYMO's specification includes optional "Hello" messages intended to maintain up to date routes and route costs.

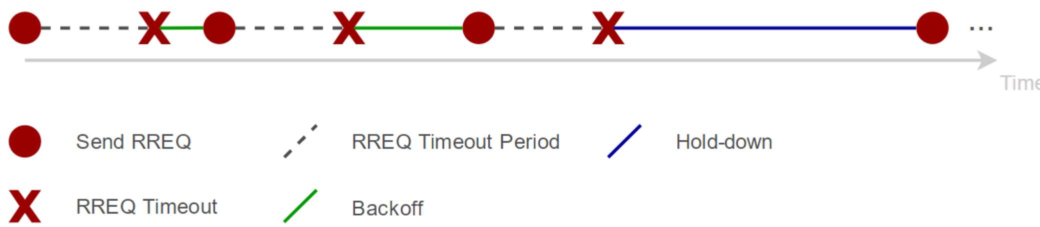


Figure 32. A node attempts three route discoveries by broadcasting an RREQ. In each case the RREQ timeout period elapses before an RREP is received. Following each timeout a back-off period is observed before sending another RREQ. Failing the maximum number of sequential discovery attempts sends a node into a hold-down state

Inactive routes in DYMO are subject to a timeout. This timeout ensures that routes which are unused for a given period are 'refreshed' before use. This refreshing process is identical to a route discovery attempt utilizing RREQs and RREPs. During route discovery attempts originator nodes utilize a timeout to handle 'lost' RREQs (Figure 32). After sending an RREQ a node will only wait a fixed amount of time for an RREP before considering the discovery attempt a failure. Multiple attempts may take place within one discovery cycle. Each failed attempt incurs a binary exponential back-off. This back-off increases the likelihood of nodes moving into a more favourable configuration and avoids repeatedly flooding the network with unnecessary RREQs. After exceeding the maximum number of failed attempts to discover a route to a target, a node will cease any further attempts and enter a "holddown" state (Figure 32). The duration of this holddown state is configurable. Holddown states only affect route discovery for 'failed' targets. The time to wait for an RREP, the maximum number of discovery attempt for a target and the duration of a holddown are all configurable.

3.2.6 Other Features

DYMO's specification includes numerous further features, several of which are not considered relevant to this work. In this section some of the less impactful features of DYMO are briefly noted.

Router Clients

All DYMO nodes discussed thus far are routers i.e. they are capable of forwarding traffic from other nodes. DYMO allows for nodes to have associated clients. Clients generate no route messages. Clients forward their packets to their "host" DYMO router. The router then carries out any necessary route discovery and forwards any client packets as required. Routers will also announce routes for their clients in a manner similar to that of an intermediate RREP. Allowing routers to have clients may reduce route message traffic and the workload of clients.

Multicast RREPs

DYMO uses multicasting in place of AODV's broadcasting. In this work all simulated network interfaces are configured to belong to the same multicast group. This configuration makes DYMO's multicast behaviour identical to that of broadcasting. This behaviour of DYMO has been ignored up to this point as there is no appreciable difference between multicasting a message to all neighbours and broadcasting the same message. For convenience, this work refers to DYMO's multicasting as broadcasting.

RREQs are broadcast by default. RREPs can also be optionally sent via broadcast rather than unicast. The primary benefit of this feature is that it may reduce the overall number of RREQs that network nodes will generate for the given target. As with RREQs, intermediate nodes which pass on RREPs will use RREPs to opportunistically create route entries.

Hop Limits

As DYMO RREQs traverse a network via broadcasting nodes update a route message hop count field. DYMO can be configured to limit the number of hops that an RREQ can make. DYMO's ability to detect RREQs which are travelling in a loop combats one aspect of wasteful route discovery. However, RREQs may continue to traverse a network even when an RREP has been generated by the RREQ's target. Hop limits combat this by allowing network designers to specify a maximum hop limit. If the longest possible, or acceptable, route within a network is *H* then any RREQ which makes more than *H* hops without reaching its target can be safely dropped.

3.2.7 D^3 Modifications

DYMO is comparatively unaltered by D^3 when compared to CubeMac and C/TDMA. D^3 introduces two inter-connected additions; The concept of a ground master (GM) and the special treatment of multiple interfaces (I/Fs). In this work's simulation, all CubeSats with the hypothetical mission's CSN have two interfaces, one for S2S and the other for S2G communications. By default, DYMO broadcasts RREQs and RERRs on the first available network interface within the appropriate multicast group. This presents an issue. If CubeSats only ever broadcast RREQs on their S2S I/F, a route to ground will never be discovered. Similarly, if only S2G I/Fs are used, then no S2S routes will be discovered.

DYMO also includes a default behaviour wherein the I/F on which a route message arrives is used as the I/F for all traffic for routes generated from that route message. As such, if a route to ground is discovered using the S2G I/F, the originator will attempt to use its S2G I/Fs for S2S communication. These issues are overcome by modifying the I/F handling behaviour of GMs. As mentioned, only CubeMac masters may participate in S2G communications. It follows that only

CubeMac masters may take the role of GM. The method used to assign the GM role is discussed in section 4.1.2.

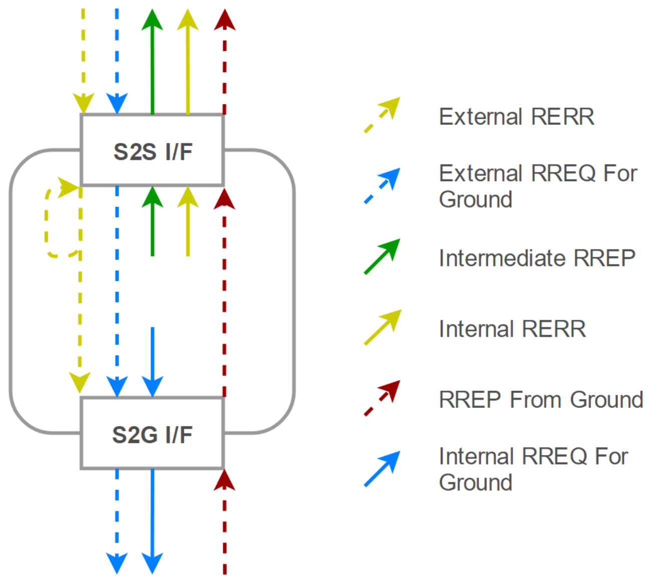


Figure 33. The interface use of a GM in response all route messages implemented by this work. In comparison, non-GMs will only ever use their S2S interfaces.

When a GM receives an RREQ for the address of ground this RREQ is sent out via the S2G I/F. Any RREP returning from ground on the S2G I/F is forwarded out via the S2S I/F (Figure 33). GMs must also change their default route storage approach to ensure that the next hop address in routes to ground references the correct I/F. Non-GMs ignore their S2G I/F and use their S2S I/F for all route messages. The final implementation of the GM role requires many small changes to the existing OMNeT++ implementation of DYMO.

3.2.8 Drawbacks

D³ has several undesirable characteristics. Most notable among these is its inability to retain redundant routes. Protocol designers must weigh the cost of a node storing redundant routes against the overall network cost of requiring a complete route discovery attempt when a route break. Within this work's hypothetical mission only routes to ground can break. This work implements the GM role such that only one master may hold the GM role at any one time. As such, all valid routes to ground will share the same final hop; from the current GM to ground. It is also assumed that nodes do not move relative to one another. Given these two assumptions there are no clear benefits in the maintenance of redundant routes as each S2G link break invalidates all possible routes to ground. Nonetheless, maintaining redundant routes may be of value in more realistic scenarios wherein the motion of CubeSats relative to one another must be considered.

As discussed in section 2.1.1, most missions will require CubeSats to periodically check the current time, and their position and velocity via GNSS receiving. Such information, when opportunistically shared amongst nodes, can greatly benefit the efficiency of route discovery, maintenance and choice. Position based routing approaches have become popular in recent VANET related research [122]. As discussed in section 4.1.2, position information is utilized in GM election. Aside from this use, D^3's lack of exploitation of readily available GNSS information represents a missed opportunity. Such information could be used to predict link breakages, avoid the use of certain unstable routes and avoid unnecessary route discovery attempts.

Designating a given master as GM adds a potentially undesirable degree of complexity to D^3 and introduces the need for distributed decision making. This work's implementation of the GM solution is presented as sufficient rather than optimal. For instance, an alternative solution may be to conduct route discovery over both interfaces from all nodes. This approach requires the use of a link cost function that reflects the difference between using an S2G I/F and S2S I/F. An appropriate cost function could reflect the difference in bandwidth and energy consumption of the two interfaces. In general, nodes would be expected to favour using their S2S I/Fs for S2S links and vice versa. However, this solution could allow an isolated node to use its S2G radio to achieve some degree of communication with the CSN to avoid total isolation from the CSN. The primary drawback of this approach is the added energy expense of conducting route discovery over multiple S2G I/Fs. Only a small number of these S2G discovery attempts will succeed.

Chapter 4: Simulation

OMNeT++ is an open-source discrete event network simulator. An OMNeT++ simulation is constructed from a number of interconnected "modules" written in C++. OMNeT++'s own "ned" syntax is used to further describe modules and define their interfaces and parameters. Modules may represent any logical or physical element within a network such as a router, software application, power supply, network connection or network protocol. Messages are passed between these modules in response to scheduled events. Messages are described using "msg" C++ template files. Through inter-module message passing OMNeT++ can be used to simulate and analyse the behaviour of a wide range of networks. OMNeT++ simulations are configured using an OMNeT++ specific configuration syntax.

Using OMNeT++, this work constructs a simulation intended to model the chosen hypothetical mission and the proposed protocols (Figure 34). Several scenarios which modify this base simulation are presented. Modules which perform the actions of CubeMac and D^3 are included in the base simulation. As far as possible, the salient operation of CubeMac and D^3 is recreated in the base simulation. Cases wherein implementations diverge from the operation of the proposed protocols, as described in chapter 3, are discussed in the following sections.

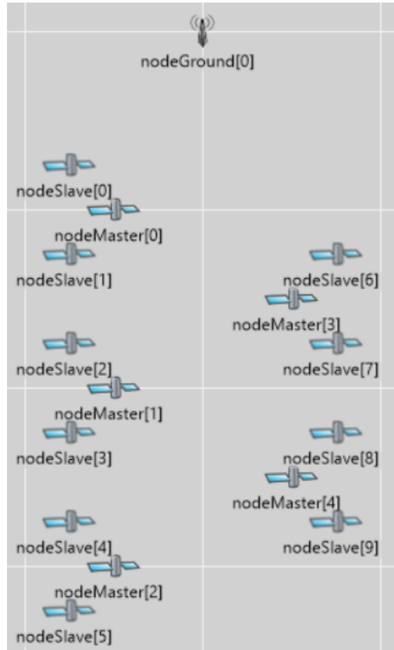


Figure 34. A visual rendering generated by OMNeT++ of the network implemented in this work's base simulation. As illustrated in Figure 3, the simulated motion of "nodeSlaves" and "nodeMasters" cause nodes to pass over ground. Video representing the network's behaviour has been made available online [123].

In order to evaluate the properties and performance of the proposed protocols a number of simulation scenarios are configured. These scenarios share several common aspects including the formation and motion of nodes as described in Figure 3 and Figure 34. Other notable scenario commonalities are as follows:

- All masters and slaves generate UDP science data packets addressed to ground.
- All network traffic utilizes IPv4 based addressing and networking concepts.
- The interval between the generation of science data packets is sampled pseudo-randomly from an exponential distribution with a rate parameter of one second.
    - The final distribution of packet generation intervals is identical across all scenarios.

- All nodes move at the same constant speed. One complete pass over the ground station takes the CSN 270 seconds. This reflects approximate orbital speeds at an altitude of 550km and reasonable S2G communication ranges.
- A pass starts with “nodeMaster[0]” at the beginning of its possible communication window with ground and ends with “nodeMaster[2]” at the end of its possible communication window with ground (Figure 34).
- Upon ending a pass all nodes immediately “wrap around” to their original position and begin a new pass.

This chapter’s remaining sections provide further detail regarding the development of this work’s simulation scenarios and the implementation of CubeMac and D^3. Details relating to OMNeT++’s operation and scenario design which are not considered impactful to the results presented in chapter 5 are omitted. All relevant materials used to develop and execute this work’s simulations have been made openly available online [124].

4.1 Implementation

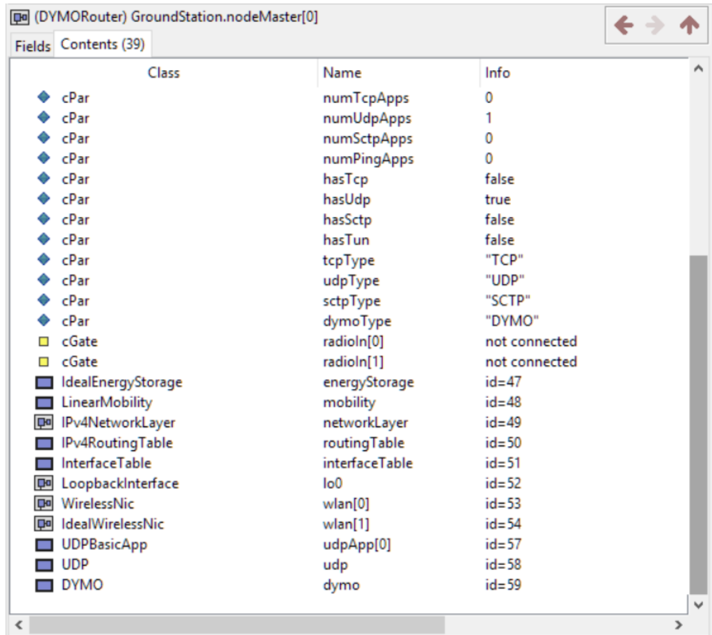
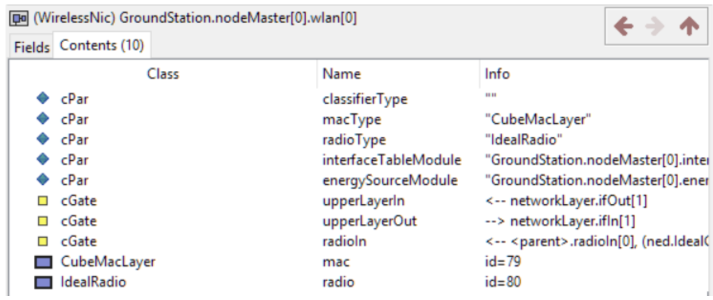


Figure 35. INET's "DYMORouter" module's various components and parameters. The module shown represents nodeMaster[0] (Figure 34). Several irrelevant parameters are omitted.

This works makes extensive use of the INET framework for OMNeT++ [125]. The INET framework is directly integrated within recent OMNeT++ releases. All relevant modules within this work's simulation scenarios are available through the INET framework, with the exception of the D^3 and CubeMac's modules. For instance, an INET module "DYMORouter" represents a network node such as nodeMaster[0] or nodeGround[0] (Figure 34). This module in turn 'contains' several INET modules which further describe the node's behaviour and properties. As illustrated in Figure 35, a module named "dymo" is contained within the DYMORouter module representing nodeMaster[0]. This DYMO module has been adapted from an existing INET module to perform the operations of D^3. Two interface modules are also present; "wlan[0]" and "wlan[1]". Wlan[0] represents nodeMaster[0]'s S2S interface and wlan[1] its S2G interface. A module which implements CubeMac is present within wlan[0] (Figure 36).


(WirelessNic) GroundStation.nodeMaster[0].wlan[0]

Fields Contents (10)

Class	Name	Info
cPar	classifierType	""
cPar	macType	"CubeMacLayer"
cPar	radioType	"IdealRadio"
cPar	interfaceTableModule	"GroundStation.nodeMaster[0].inte
cPar	energySourceModule	"GroundStation.nodeMaster[0].ener
cGate	upperLayerIn	<-- networkLayer.ifOut[1]
cGate	upperLayerOut	--> networkLayer.ifIn[1]
cGate	radioIn	<-- <parent>.radioIn[0], (ned.Ideal(
CubeMacLayer	mac	id=79
IdealRadio	radio	id=80

Figure 36. The parameters of and modules contained within nodeMaster[0]'s "wlan[0]" module.

In response to a scheduled event, nodeMaster[0]'s "UDPBasicApp" module will generate a 128 Byte UDP packet addressed to nodeGround[0]. This packet represents a science data packet. This packet is passed to the "IPv4NetworkLayer" module. In the case that no route to ground can be found by the "IPv4RoutingTable" module the packet will be passed to the "DYMO" module, which implements D^3. D^3 will enqueue this packet pending a successful route discovery for the address of nodeGround[0]. Once a route has been established to nodeGround[0], the packet will be removed from the D^3's internal queue and sent to the appropriate interface (Figure 33). Assuming the S2S interface is to be used, the packet will be passed to the "CubeMacLayer" module within the "wlan[0]" module. CubeMac will enqueue the packet pending the start of nodeMaster[0]'s time slot. During this time slot CubeMac will send the packet to the "IdealRadio" module which will in turn sends the packet to a module named "IdealTransmitter". From this point the packet will pass through an "IdealRadioMedium" module. Finally, the packet will be received by an "IdealReceiver" module within the S2S interface of another network node. The packet will be passed from the receiver up through a similar progression of modules which will determine the receiving node's handling of the packet. The module path taken by this packet is illustrated in Figure 37.

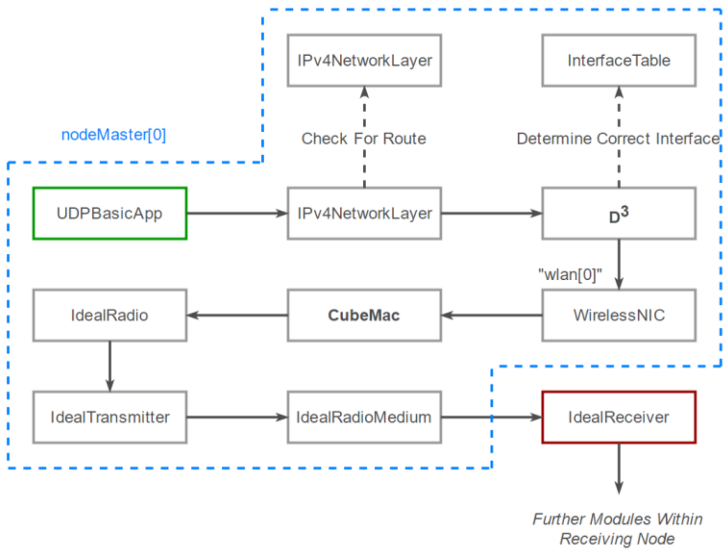


Figure 37. An example packet's progression through various modules within nodeMaster[0] which culminates in its reception at another node within the CSN. This path represents a common progression including both D^3 and CubeMac.

This work's simulation utilizes several "Ideal" INET modules. As discussed in chapter 3, aspects of the physical layer are not considered relevant to the analysis of D^3 and CubeMac. Idealized modules allow for greatly simplified simulations wherein the effects of modifying modules or adjusting parameters may be more easily understood. The largest drawback resulting from the use of ideal modules is the lack of an accurate model of CubeSat communication within LEO environments. For instance, the effects of noise, propagation delays, signal interference and signal fading are not modelled. The use of ideal models was driven by the need for rapid prototyping and several issues arising from the module modifications required by CubeMac and D^3. Such issues are discussed further in section 4.2.

4.1.1 CubeMac

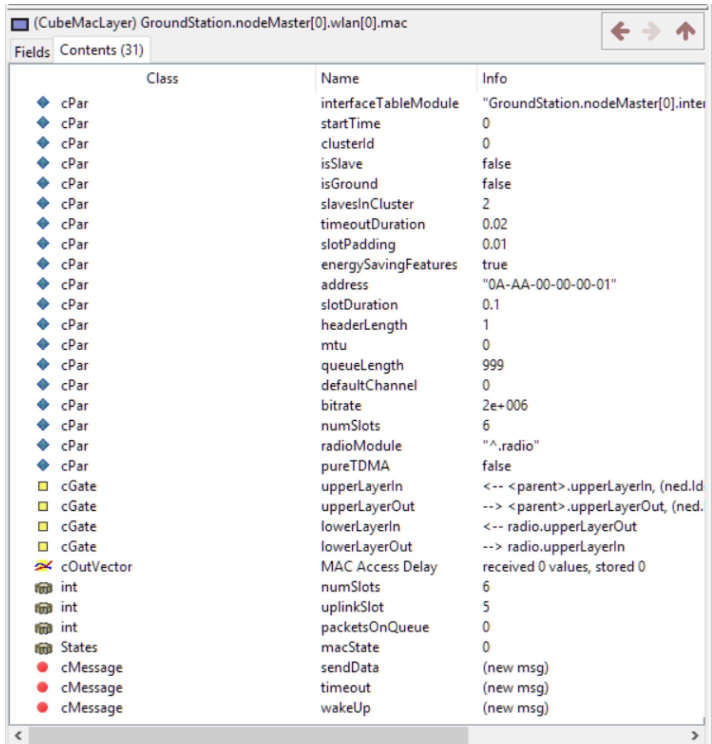


Figure 38. The various visible elements of a CubeMac module. From top to bottom are: Parameters, gates (connection points between modules), a vector (for result recording), watched internal module variables and owned messages. The above messages are all used for scheduling internal events.

The CubeMac module was developed following the design approach of an existing INET implementation of LMAC [119]. Internally CubeMac implements two large state machines, one for CubeMac slaves and the other for CubeMac masters. The CubeMac module's implementation is highly faithful to the description of CubeMac in section 3.1.

CubeMac can be configured to act in a pure TDMA mode using the "pureTDMA" parameter. In this mode, all nodes act as masters which removes all clustering and CDMA. The behaviour of D^3 is unchanged by this mode. Although the slave role is dropped, only nodes originally designated as masters (nodeMasters) may conduct S2G communication. CubeMac's default operation is compared to its pure TDMA mode in simulation scenario 2a.

C/TDMA's time slots are modelled through tracking the ID of the current time slot. Internally a 'self-message', "wakeUp", is scheduled at intervals equal to the configured slot duration. Upon receiving a wakeUp message nodes will determine the current slot and carry out any required actions. For masters, if the starting slot ID matches its cluster ID parameter the master behaves as an owner of the time slot. A similar approach is used for the slave uplink slot wherein all slaves know the ID of the uplink slot.

CDMA's implementation is trivial. All slaves enter a "send data" state during the uplink slot and send packets to the IdealRadio module (Figure 37). The parallel communication of CDMA required adjustments to the INET "Radio" module which is a parent module of the IdealRadio module. These adjustments are discussed in section 4.2.

4.1.2 D^3

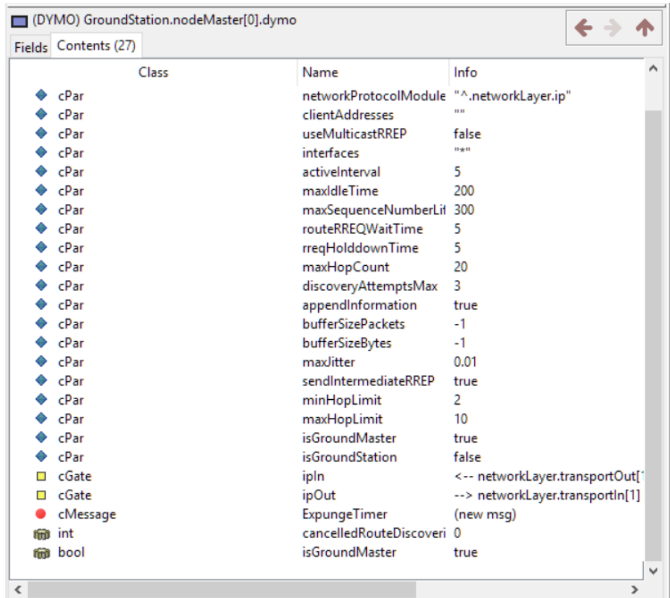


Figure 39. Parameters such as "isGroundMaster" relate to D^3's modifications of DYMO as described in section 3.2.

D^3 retains the majority of the existing INET implementation of DYMO. Any references to a DYMO module in simulation scenarios should be taken to refer to the D^3 protocol. D^3 implements two new parameters: "isGroundMaster" and "isGroundStation". These parameters are used to handle the modifications discussed in section 3.2.7. D^3 is implemented in a more functional manner than CubeMac. D^3 specifies dedicated functions for handling each type of route message. D^3's current state is almost entirely described by the route entries in the containing node's IPv4RoutingTable module.

Several aspects of the relevant IETF DYMO specification [83] were found to be either omitted or incorrectly implemented within the existing INET DYMO module. RERRs were non-functional, link costs and sequence numbers were incorrectly implemented and no mechanisms for loop detection were present. Of these, link costs, sequence numbers and loop detection were made operative in D^3. A 'stand-in' solution for RERRs was implemented alongside the logic for the election of ground masters.

As discussed, DYMO's approach to interface handling is altered in D^3 to match the behaviour of GMs and non-GMs. The implementation of GM election constitutes the greatest departure from a viable "real-world" implementation within the simulation of proposed protocols. GM election requires information regarding multiple nodes. For instance, election of the master closest to the ground station requires knowledge of the current location of all masters. Rather than a distributed decision making approach based on information shared in packet headers, an additional "RoleOracle" module is added to the base simulation.

D^3's role oracle module periodically "wake ups" at configurable intervals in order to determine whether the GM role should transfer to a new master. Using the "closest master" approach, the oracle will traverse the module hierarchy using built-in OMNeT++ functions. During this traversal the oracle collects the positions of all masters and determines which master is currently the GM. If the current GM is not the master closest to ground the oracle will call custom functions to transfer the GM role to the closest master (Figure 40). Simulation scenarios are configured such that at least one master is always within communication range of ground. As a result, one master will always be designated as the GM. Also, each transferral of the GM role constitutes the only possible source of a link break. Typically, this would require the master vacating the GM role to broadcast a RERR. However, as discussed, the implementation of RERRs in the existing INET DYMO module was found to be fundamentally broken. In place of RERRs propagating throughout the network, the oracle calls a function for each CSN node. This function instructs the node to drop all routes to ground and cancel any ongoing route discovery attempts.

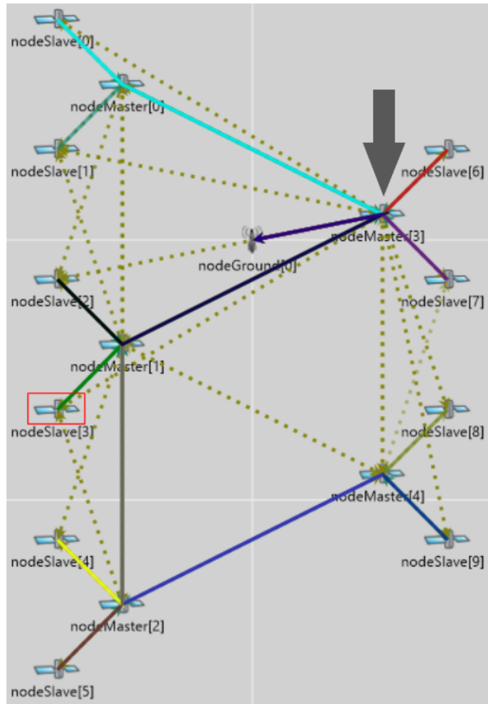


Figure 40. The closest master to ground, indicated by the grey arrow, is elected as the CSN's GM. The paths of UDP packets are indicated by solid lines. Logical routes determined opportunistically as a result of the movement of route messages are indicated by dotted lines.

Figure 40 indicates the use of a potentially unexpected route to ground by nodeMaster[4]. Rather than performing a single hop to nodeMaster[3], nodeMaster[4] opts for a route involving three hops starting with nodeMaster[2]. There are several possible causes of this route choice. Rather than a hop count based link cost function (Figure 31) D^3 implements a cost function based on a receiving node's energy consumption since the beginning of simulation. This link cost allows D^3 to favour routes involving nodes which have consumed the least amount of energy. However, this cannot be the cause of the route choice in Figure 40 as the available route to nodeMaster[3] is necessarily cheaper (All link costs must be values greater than zero). The most likely cause of this route choice is an unresolved bug within INET's DYMO module which has been carried into D^3. In this case, nodeMaster[4] began route discovery and received an intermediate RREP from nodeMaster[2]. An RREP from ground or nodeMaster[3] should also have reached nodeMaster[4] and allowed it replace its more expensive existing route to ground. It is possible that D^3's use of sequences numbers caused nodeMaster[4] to drop the cheapest route to ground in favour of a 'fresher' route represented by the intermediate RREP from nodeMaster[2].

4.1.3 Parameterization

This work's simulations are configured through several hundred parameters. Such parameters, as shown in figures such as Figure 36, control various aspects of operation of simulation modules. The parametrization of node positions results in an in-cluster spacing of ~35m and the shortest inter-master spacing of ~112m. Such distances are not to scale and are used for ease of visualization. The use of IdealRadio modules negates all realistic communication distance effects other than the establishment of whether two radios are in within wireless communication range. Further relevant parameters are listed in Table 2.

Table 2. Parameter values are based, where possible and practical, on the known capabilities of CubeSats.

Module	**Parameter**	**Value**	**Note**
UDPBasicApp	sendInterval	Exponential(1s)	Section 4.1
CubeMac	slotDuration	100ms	[111]
CubeMac	*	*	Figure 38
DYMO	*	*	Figure 39
IdealTransmitter (S2S)	communicationRange	125m	Unrealistic. Adjusted to improve visualization
IdealTransmitter (S2G)	communicationRange	150m	Scaled from 750km

LinearMobility	intialZ	100m	Scaled from 550km
LinearMobility	Speed	1.5m/s	Produces a realistic S2G contact duration
S2G I/Fs	Bitrate	125kbps	[40]
S2S I/Fs	Bitrate	2Mbps	Gamalink, section 2.1.1
Master S2S I/Fs	receivingPowerConsumption	200mW	Gamalink, section 2.1.1
Master S2S I/Fs	transmittingPowerConsumption	1.5W	Gamalink, section 2.1.1
Slave S2S I/Fs	receivingPowerConsumption	225mW	+16% For CDMA (Assumption)
Slave S2S I/Fs	transmittingPowerConsumption	1.75W	+16% For CDMA (Assumption)
S2G I/Fs	receivingPowerConsumption	500mW	Assumption
S2G I/Fs	transmittingPowerConsumption	3W	TW-1 mission, section 2.1.1
All I/Fs	sleepPowerConsumption	1mW	Assumption
All I/Fs	switchingPowerConsumption	2mW	Assumption

4.1.4 Scenario 1a & 1b

Scenario 1a acts as the baseline scenario. This scenario represents the base simulation as discussed in the preceding sections. Scenario 1b is a slight modification of Scenario 1a wherein certain the additional CubeMac energy saving features are disabled. Specifically, "No Data" and "Last" packets are disabled. The only configuration change made by Scenario 1b is setting CubeMac's "energySavingFeatures" parameter to "false".

4.1.5 Scenario 2a & 2b

Scenario 2a and 2b are used to compare the performance of CubeMac with its pure TDMA mode and an existing INET CSMA MAC protocol respectively. In scenario 2a, CubeMac has its Boolean "pureTDMA" parameter set to "true" (Figure 38). All slaves have their parameters adjusted to match those of master nodes.

In scenario 2b, the CubeMac module in S2S I/F modules is replaced with an existing INET CSMA module. This module is configured not to use acknowledgements and where possible to match the corresponding parameters of CubeMac in scenario 1a.

4.1.6 Scenario 3

The default "closest-master" approach is used by the RoleOracle in scenario 1a. In scenario 3, the oracle is configured to utilize an approach which considers a master's distance from ground as well as the amount of energy consumed by the master since the start of a simulation run. The "energy-distance" approach calculates a score for each master. The master with the lowest score is considered to be the most favourable GM. Each score is calculated as:

$$Score = d_G + \left(d_G * (ER * W_{ER})\right)$$

$$d_G\text{: } Distance\ to\ Ground - \ ER\text{: } Energy\ Rank - W_{ER}\text{: } Energy\ Rank\ Weight$$

Energy ranks are calculated as: 1 + the number of masters which have consumed less energy overall than the master in question. A rank of '2' indicates one master exists which has consumed less energy than the master in question. The energy rank weight is a value passed to the oracle as a parameter. It is used to tune the impact of a master's energy rank on its final score. A weight of zero will result in election identical to that of “closest-master”. Following experimentation, a final weight of 0.3 was selected.

This approach is inspired by the Nodes mission’s captaincy election. It is worth noting that this election approach could be further extended to include election factors considered by Nodes CubeSats i.e. battery voltage, the amount of science data collection and the predicted duration of the next ground pass.

4.2 Issues

The development of simulation scenarios faced numerous issues, some of which could not be fully addressed given this work’s available development resources. Such resources were reduced as a result of an early change in project direction as well as a change in simulation tool. Development began with Network Simulator 3 (NS-3) [126]. However, it became apparent that many aspects of the hypothetical mission and proposed protocols would have to be written from scratch for NS-3. OMNeT++, with its inclusion of the INET framework, was determined to be a more suitable tool for this work’s purposes.

4.2.1 CDMA

Simulations are configured to use INET’s IdealRadio modules (Figure 37). These modules inherit the majority of their code from the INET “Radio” module. The Radio module is extended and accessed by numerous other INET modules. The Radio module presented an issue as it is explicitly designed to handle one packet reception at a time. Upon receiving a packet as a “reception” the Radio module schedules a “receptionTimer” which generates an event at the calculated end of a packet reception. In handling this event the Radio module determines whether the reception was successful. Only one receptionTimer self-message is implemented within the INET Radio module. This presents an issue as CDMA requires the handling of multiple concurrent receptions. If two packets arrive at once, or if a packet arrives before the 'end' of an ongoing reception, only one reception will succeed due to reuse of the receptionTimer message. This failure to receive a packet occurs silently. As such, correctly identifying the root cause of this issue consumed considerable resources.

This issue was resolved through numerous modifications to the Radio module which allowed the module to create a receptionTimer for each incoming packet. As mentioned the Radio module is associated with many other INET modules. The modifications required to resolve this issue broke several realistic radio modules based on the Radio module. This forced a development choice; continue attempting to modify the INET modules representing more realistic radios or continue with "ideal" modules.

4.2.2 Routing Protocol Modules

The INET framework provides several routing protocol modules based on well-known protocols such as AODV [87] and GPSR [94]. Of these, AODV was this work's first choice for use alongside CubeMac. However, AODV, and several other routing modules, failed to perform as expected. In the case of AODV, the INET implementation requires several parameters which estimate the time taken for route message to traverse a given network. Due to the asymmetrical nature of CubeMac based communication even highly accurate estimates resulted in numerous instances of packet loss, cycles of route discovery failure and runtime errors. Although it was found that, through experimentation, AODV timing parameters could be tuned to reduce ill effects, this approach proved highly impractical as development progressed.

DYMO was used in place of AODV as it did not rely on such timing parameters. However, the INET DYMO module did not work as expected immediately either. The module could eventually establish routes but, due to a lack of loop detection, would generate several thousand route messages during discovery attempts.

4.2.3 DYMO

As discussed in section 4.1.2, INET's DYMO module included various bugs and omissions relating to route metrics, loop detection and RERRs. Aside from these, two other major issues consumed considerable development resources. These issues related to multiple interface handling and DYMO's use of multicast. The former issue is introduced in section 3.2.7. Section 3.2/7 omits several issues which had to be addressed in order to implement the desired multiple interface behaviour (Figure 33).

By default, DYMO is implemented to announce, a single "routerID". In many cases, this ID is added to an address field of a handled route messages. In the case of this work, this ID is an IPv4 address. This address is read from the first I/F to be configured during OMNeT++ module initialization phases. However, the use of the addresses of both I/Fs is required. Several workarounds were necessary to ensure the address of the correct interface was included in route messages. For instance, if the address of an intermediate node's S2S I/F was placed in an RREQ sent via the S2G I/F, this would result in ground recording an incorrect next hop address in a route entry for this node. Packets that use this route entry would be incorrectly ignored by the intermediate node's S2G I/F.

The DYMO module also makes several routing table related function calls which return a 'default' interface or the interface referenced by the first matching entry in the "InterfaceTable" module.

Such calls often resulted in an incorrect interface being used for communications. In short, the existing INET DYMO module was found to be poorly suited for multiple interface use cases.

Within the DYMO module, the use of IPv4 multicast was correctly instrumented. However, several modules used in this work's simulations either could not initially handle IPv4 multicast addressing or did not behave in the manner required by DYMO. This issue was addressed by registering all interfaces within the same multicast group and modifying several modules to ensure correct handling of IPv4 multicast addressing.

4.2.4 Intermittent Failures

The issues discussed thus far represent the most notable issues for which the root causes were identified. Several other issues arose during development the root causes for which could not be identified. Although it was possible to avoid many these issues, one issue arose near the end of the development of simulation scenarios. As OMNeT++ exceeds approximately one million simulated events the likelihood of a seemingly random segmentation fault increases. These faults greatly hindered the development of simulation scenarios. Despite the dedication of considerable resources to attempting to understand and debug these faults no viable root cause could be identified. The faults exhibit no clear pattern other than becoming more likely as the number of simulation events increases.

Chapter 5: Results

This chapter presents and discusses results collected from this work's simulation scenarios. Each of the five scenarios is run for 810 seconds of simulation time. This 810 second period represents three consecutive passes over ground as described in the opening section of chapter 4. OMNeT++ is configured to record numerous network performance metrics during each scenario run. These metrics are used to compare the performance of the various simulation scenarios. In this chapter, two core metrics are used to analyse the performance of the proposed protocols with respect to the PvTP trade-off: the number of packets received by the ground station and the total amount of energy consumed by each node.

All data packets have a fixed size of 128 Bytes. As such, only the number of packets received, and not the quantity of data, is of consequence. Also, both the total number of packets generated (12234) and the distribution of packet generation are fixed across scenario runs. The distribution of packet generation is such that each node will generate a packet, on average, once per second. This produces an overall packet generation rate of approximately 15 packets per second. Based on the available S2G bandwidth of 125kbps, the maximum theoretical rate of packet reception at ground is 16 packets per second. The packet generation rate is lowered below this 16 packets per second saturation point to accommodate the effects of ground master election and on-demand route discovery.

D^3 route messages are excluded from received packet counts. However, the effects of D^3 activities on energy consumption are not omitted. Simulated node radios provide the only sources of energy consumption within simulation scenarios. The salient parameters for radio modules and submodules are presented in Table 2. Simulated nodes do not recharge their energy stores as they would in reality. As all scenarios utilize the same base orbital parameters, the recharge experienced by nodes would be identical across scenarios and therefore would produce no effect on scenario comparisons.

5.1 Scenario 1a

As discussed in section 4.1.4, scenario 1a represents this work's baseline scenario. D^3 and CubeMac are utilized in their default states as described in chapter 3. D^3's oracle is using the aforementioned "closest-master" election approach and CubeMac has all additional energy saving features enabled.

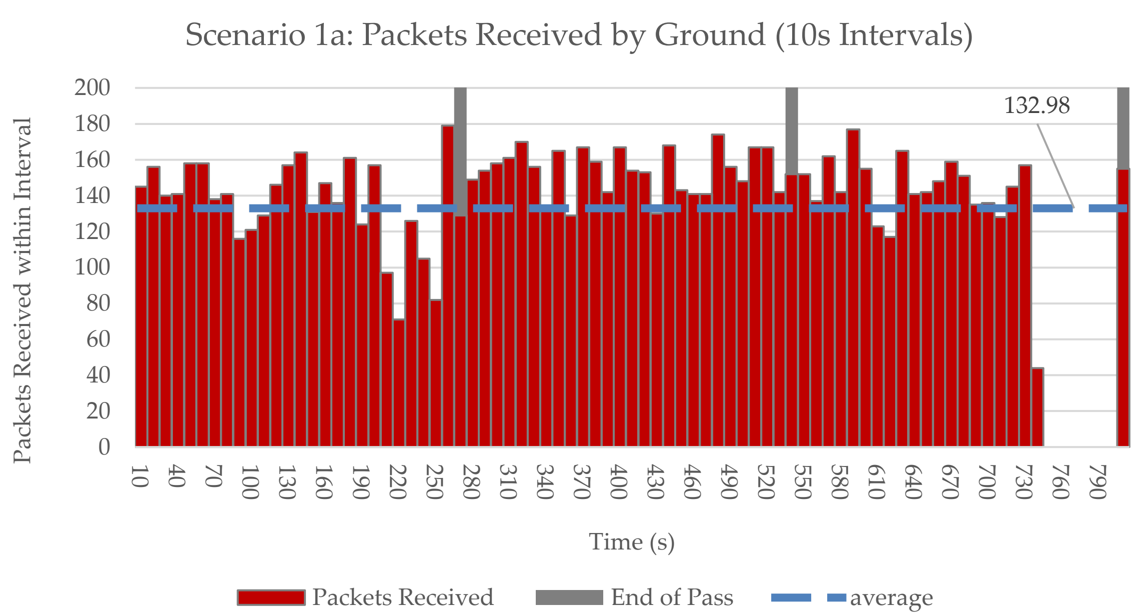


Figure 41. The number of packets received by ground station over time reduced to a granularity of ten second intervals. Each pass lasts 270s. The end of each pass is represented by a grey column.

An anomaly is present in the results shown in Figure 41. A drop in packet reception can be seen during periods when 'nodeMaster[2]' occupies the GM role in the first and last ground passes. NodeMaster[2] is the last master in the CSN to obtain the GM role during each pass (Figure 40). NodeMaster[2] holds the GM role for ~85 seconds beginning at the 185 second mark of each pass (Table 3). During the first pass there are notable drops in the number of packets received at ground while nodeMaster[2] is the GM (185s – 270s) (Figure 41). While nodeMaster[2] is the GM during the third and final pass (725s – 810s), packet reception drops to zero for 60 seconds (Figure 41).

Table 3. Each master's start time as GM and the duration spent as GM during a single pass. These times reflect the closest-master default election approach of D^3's oracle and the physical layout of nodes (Figure 40).

Master	GM Start Time	GM Duration
0	0s	85s
3	85s	35s
1	120s	30s
4	150s	35s
2	185s	85s

These packet reception drops are not expected features of D^3 and CubeMac's combined behaviour. Neither protocol module reported unexpected states or erroneous behaviour during these periods. Also, the second pass shows no signs of the anomaly. The root cause of this anomaly could not be identified. Considering the various issues discussed in section 4.2, it is possible that changes made to core INET modules caused this anomaly. The probability that these changes being are the cause of this anomaly is strengthened by the fact that this anomaly does not affect scenario 2a's results. Scenario 2a uses no CDMA based communication which, as discussed in section 4.2.1, relies on several modifications to core INET modules.

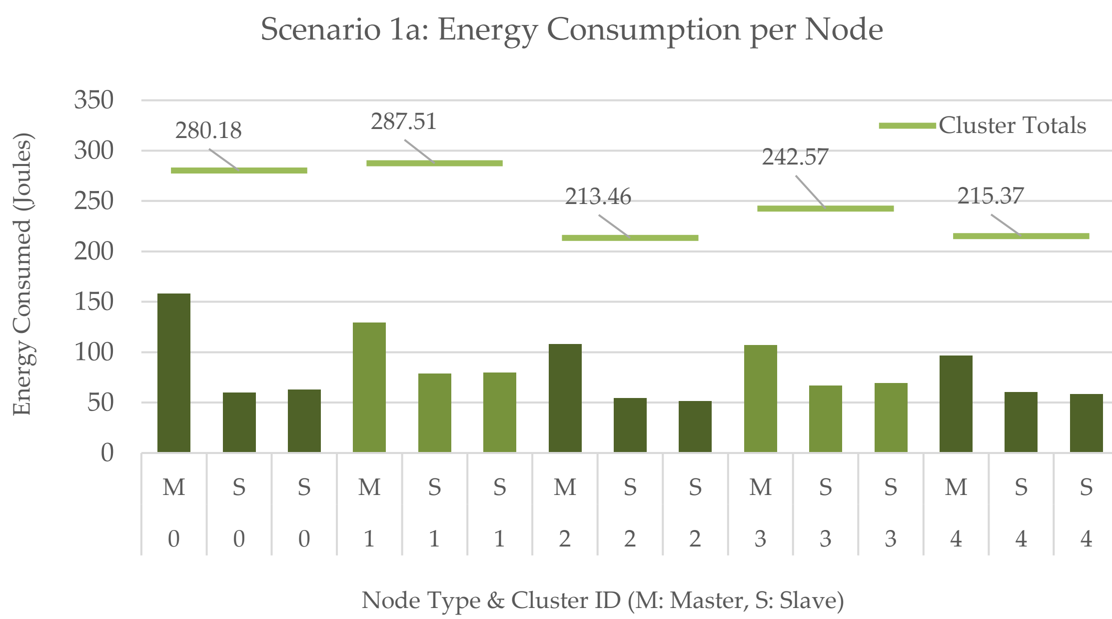


Figure 42. The total energy consumed per node during scenario 1a's simulation run. As masters must handle the routing of all slave packets as well as S2G communications, their energy consumption is notably higher. Masters consumed an average of ~120J and slaves consumed an average of ~64J.

As expected from the design of CubeMac, masters consume more energy overall than slaves. It is also evident from Figure 42 that the master of cluster 0 (nodeMaster[0]) consumes the most energy overall. This is expected as this master holds the ground master role for the longest possible duration under closest-master election (Table 3). Without the observed anomaly, it is expected that nodeMaster[2] would also consume a similar amount of energy as nodeMaster[0].

Packet generation intervals are established through the pseudo-random sampling of an exponential distribution as described in section 4.1, resulting in an uneven distribution of the total number of packets generated per node. This is the likely cause of fluctuations in slave energy consumption. Within a single cluster slave energy consumption results are closely matched.

It is expected that all nodes within cluster 1 would experience higher levels of energy consumption as a result of the cluster being the most central of the CSN (Figure 40). This centrality results in the cluster handling a greater number of D^3 route messages which increases intra-cluster communication. The impact of this centrality is clear in the energy consumption of cluster 1's master. NodeMaster[1] occupies the GM role for the shortest amount of time per pass (30s). Despite this, nodeMaster[1] has the second highest level of energy consumption.

5.2 Scenario 1b

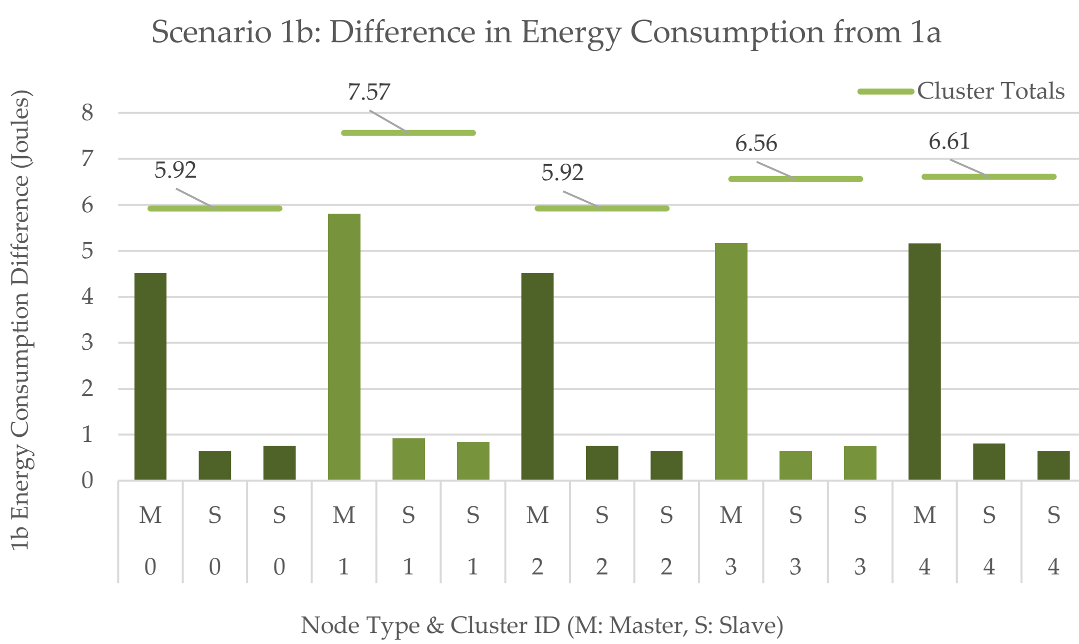


Figure 43. Energy differences calculated as the energy consumed by each node in scenario 1b less the energy consumption of corresponding nodes in scenario 1a.

CubeMac augments C/TDMA with additional energy saving features. In scenario 1b these features are turned off in order to test their impact. All other aspects of scenario 1a remain fixed. Scenario 1b also suffers from the aforementioned nodeMaster[2] anomaly. Scenario 1b's packet reception results are identical to those of scenario 1a (Figure 41) therefore these results are not shown. The lack of change in packet reception illustrates that the added "no data" packets and "last" packet fields have no impact on S2G throughput. Figure 42 illustrates the change in energy consumption per node in scenario 1b as compared with scenario 1a. All nodes show higher levels of energy consumption without CubeMac's additional energy saving features. Variation between energy consumption differences are low across both master and slave nodes. NodeMaster[1]'s higher change in energy consumption may be a result of its centrality. If so, this suggests that the performance of CubeMac's energy saving features increases with a node's traffic workload. On average, masters and slaves consumed ~4.2% and ~1.2% more energy respectively for an overall increase in energy consumption of ~2.6% in scenario 1b.

5.3 Scenario 2a

CubeMac's pure TDMA mode removes the concept of node clusters and the use of CDMA by assigning all nodes to the master role. This change requires that the dedicated slave uplink slot be dropped from the CubeMac frame. The total number of master node slots must then extend from five to fifteen. Scenario 2a is parameterized such that nodes which were previously slaves exhibit the behaviour as masters. All other aspects of CubeMac and scenario 1a are unchanged. D^3's behaviour is also unchanged. As such, D^3 will only elect GMs from the original group of masters.

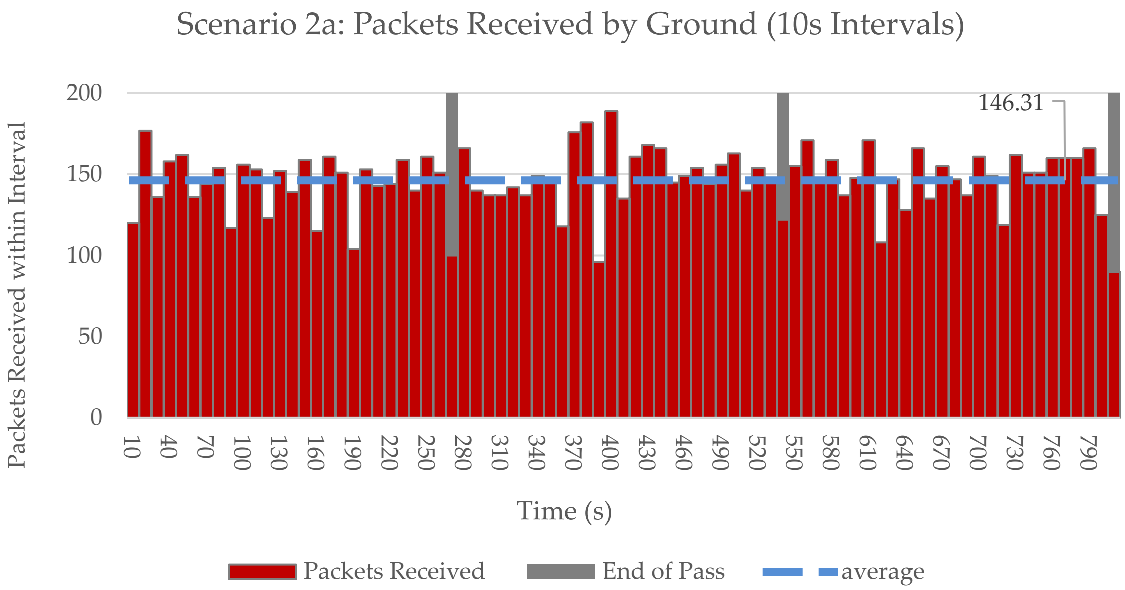


Figure 44. When compared with Figure 41 CubeMac's pure TDMA mode can be seen to provide more consistent throughput.

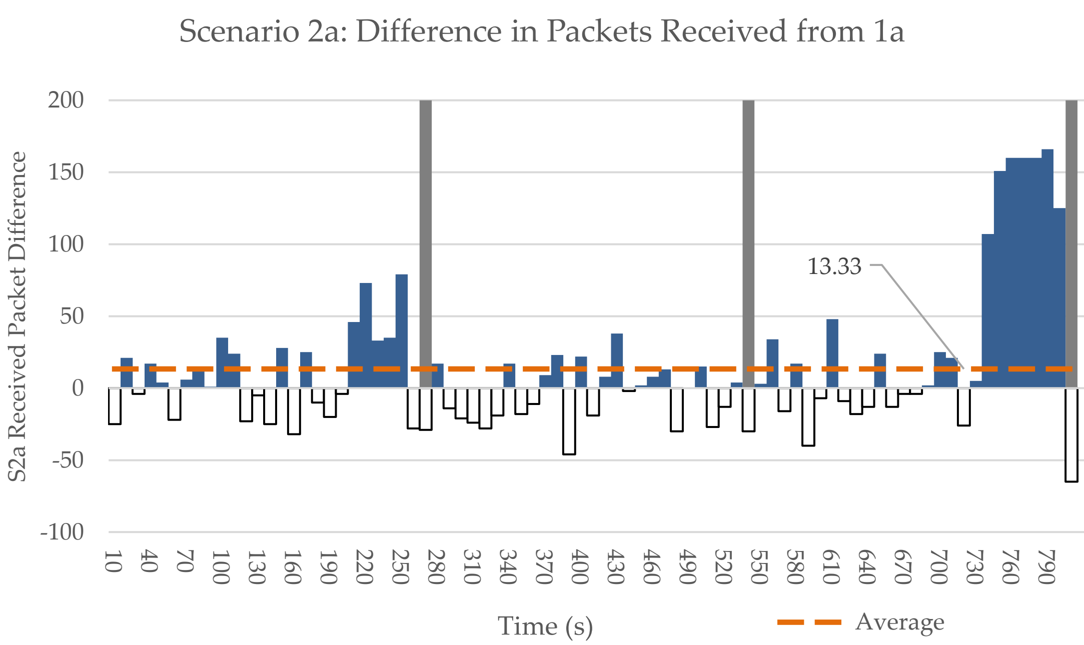


Figure 45. The difference over time in the number of packets received in scenario 2a as compared with scenario 1a. Negative values (outlined) represent interval values in which scenario 1a's ground station received more packets than scenario 2a's ground station.

It is evident from Figure 44 and Figure 45 that pure TDMA does not suffer from the scenario 1a anomaly. This suggests the anomaly is related in some way to the behaviour of CubeMac slaves. Scenario 2a has a total packet reception result of 11851 which is an increase in 1080 packets over scenario 1a. After adjusting for the scenario 1a anomaly, this increase falls to ~100 packets. This represents an increase of less than 1% over scenario 1a. The statistical significance of this adjusted result is dubious. The examination of the true S2G throughput performance difference between scenario 1a and 2a requires the resolution of scenario 1a's anomaly.

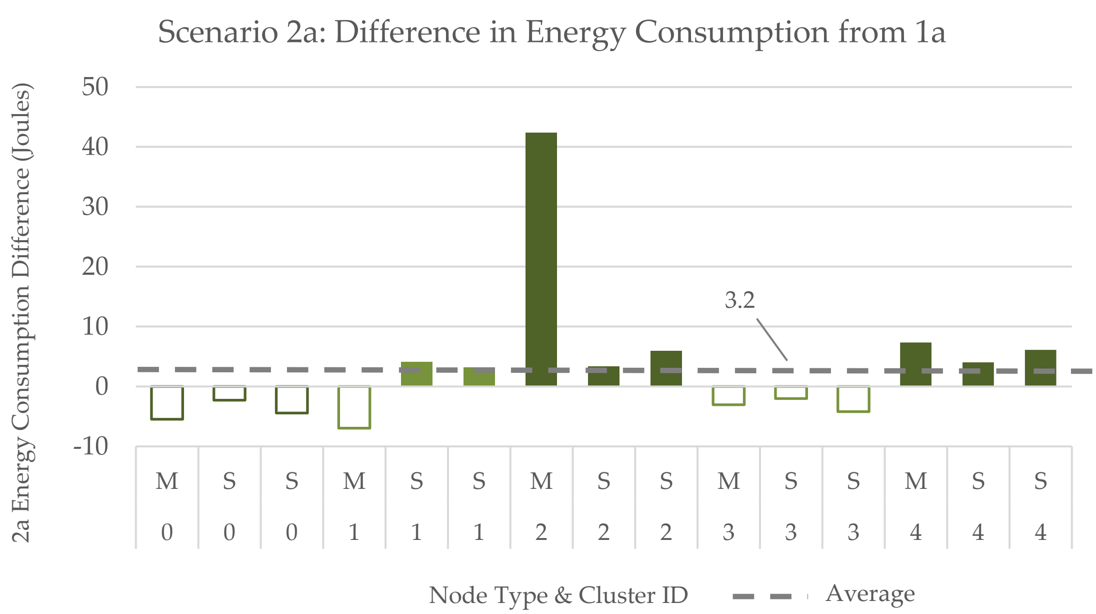


Figure 46. The change in energy consumption of individual nodes in scenario 2a as compared to scenario 1a. Negative (outlined) columns represent instances in which nodes in scenario 2a, consumed less energy overall than corresponding nodes in scenario 1a.

As with packet reception results, differences in scenario 2a's energy consumption results are effectively nullified by scenario 1a's anomaly. As expected, Figure 46 shows that nodeMaster[2]'s energy consumption increases significantly when it communicates with ground for the expected 255 second duration (Table 3). All other changes in node energy consumption are likely to be the result of the greater number of possible routes that may be formed as a result of pure TDMA's removal of CubeMac's default cluster formation (Figure 47).

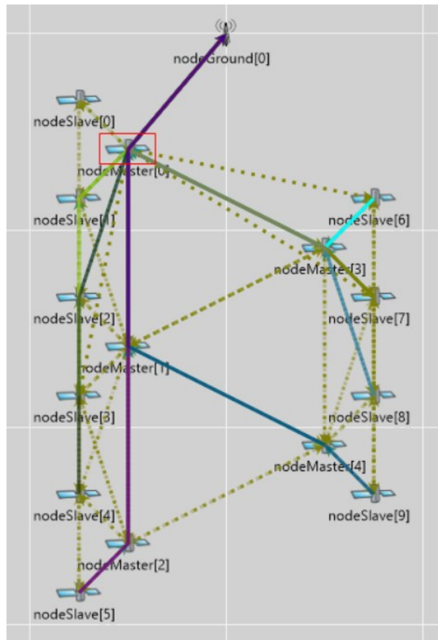


Figure 47. In CubeMac's pure TDMA mode all nodes act as masters. This arrangement creates a larger number of viable routes to ground. Solid lines above represent the movement of science data packets. This behaviour can be compared to the default CubeMac & D^3 behaviour as illustrated in Figure 40.

Scenario 2a's performance, in terms of energy consumption and S2G throughput, approaches equivalency with that of section 1a when results are adjusted for section 1a's anomaly. However, scenario 2a's performance in terms of the timeliness of packets received at ground is significantly removed from that of scenario 1a. Timeliness, in this case, may be measured in terms of packet "end-to-end" (E2E) delays. A packet's E2E delay is calculated as the amount of time taken for the packet to reach ground following its generation by a "UDPBasicApp" module (Figure 37).

Time slots have a fixed duration of 100ms. Thus, each CubeMac frame has a duration of 600ms in scenario 1a and 1.5s in scenario 2a. These frame lengths are a central factor in the E2E delays. The average packet E2E delay for scenario 1a is 676ms whereas the average E2E delay in scenario 2a is 1.86s. Although scenario 2a achieves similar performance in terms of throughput and energy consumption, each packet in scenario 2a takes on average 1.184s longer to reach ground than in scenario 1a. This is a direct result of scenario 2a's longer CubeMac frame. Due to the TDMA nature of CubeMac, packets which must make *H* hops to reach ground may experience worst case delays of approximately *H-1* frame durations provided each CSN node has no existing packets queued. Section 3.1.6 discusses possible worst-case packet delays which can occur within a CubeMac frame. Figure 27 illustrates ill-timed packet generation.

5.4 Scenario 2b

In scenario 2b the custom CubeMac module is replaced with an existing INET module which implements a CSMA based protocol. As in scenario 2a, as few elements of scenario 1a are altered as possible. Where possible, the parameters of the utilized CSMA module are adjusted to match those of CubeMac in scenario 1a.

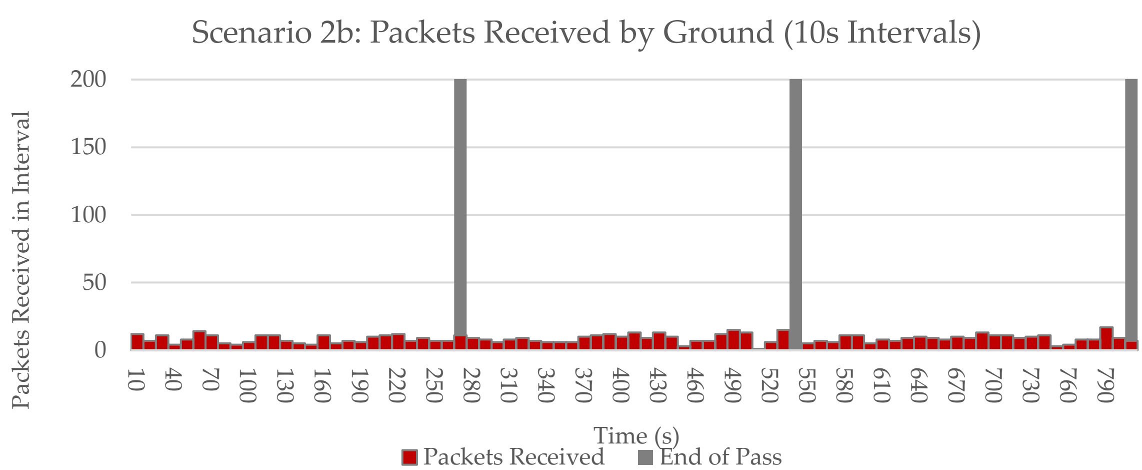


Figure 48. This figure can be compared to both Figure 41 and Figure 44 which use the same y-axis scale.

Figure 48 shows an order of magnitude decrease as a result of the use of CSMA in place of CubeMac. This approach results in over six times the amount of energy being required to send a single packet to ground in comparison to scenario 1a (Table 4). Radhakrishnan et al. report an average inter-node throughput of 24% from simulations of a CSMA based protocol. Radhakrishnan et al. compare this result to C/TDMA's 95% inter-node throughput performance [111]. In this work, scenario 1a achieves an S2G throughput of ~88%. This throughput percentage is measured as the total number of packets received at ground divided by the total number of packets generated. In comparison, scenario 2b achieves an S2G throughput of ~6%. From the work of Radhakrishnan et al. it is expected that CubeMac will out-perform a contention based MAC protocol based on CSMA. However, the considerable gap between scenario 1a's and scenarios 2b's performance calls into question the fidelity of scenario 2b's results.

Considering the number and frequency of INET related issues encountered by this work, it is feasible that the utilized CSMA module contains fundamental flaws. Lacking a thorough investigation and validation of this module's implementation, the accuracy of scenario 2b's results cannot be guaranteed. However, the low packet reception rates may be reflective of the combination of CSMA with D^3.

Route discovery in D^3 relies on flooding the CSN with RREQs through a series of broadcasts. The D^3 oracle's instantaneous removal of all routes to ground in response to a change in GM causes nodes to initiate route discovery attempts. Simultaneous route discovery attempts by all nodes will result in numerous collisions which will slow CSMA based communication. This slowing of CSMA based communication will result in an overall increase in the time required to discovery routes to ground. Slowing route discovery increases the chances of node timing out a discovery attempt and observing a back off period (Figure 32). Overall, the use of CSMA greatly reduces the time taken for nodes to obtain valid routes to ground thus reducing the effective time available for a node to communicate its packets to the current GM. This, in turn, reduces maximum possible data packet reception at ground.

5.5 Scenario 3

D³'s default approach to GM election is to elect the closest master to ground at any given time. Scenario 3 presents the effects of an alternate approach which utilizes both a master's distance to ground and the energy consumed by a master. This approach is described in detail in section 4.1.6. Aside from an alternative GM election approach scenario 3 is identical to scenario 1a.

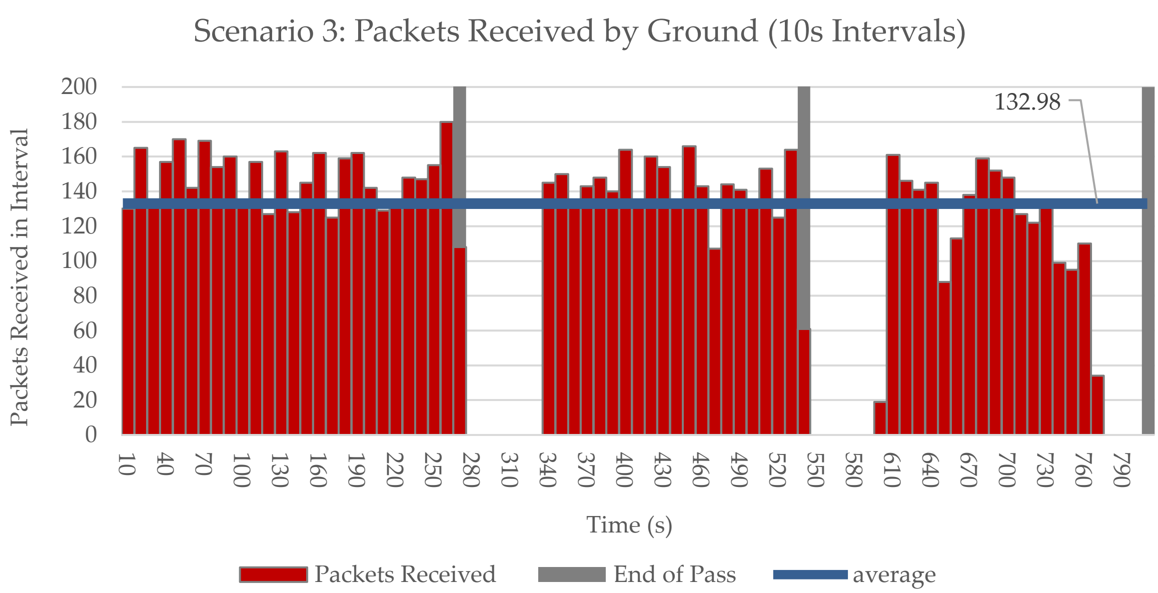


Figure 49. Scenario 3 imposes greater restrictions on access to ground through an adjusted ground master election approach. This approach results in several deliberate "gaps" in packet reception.

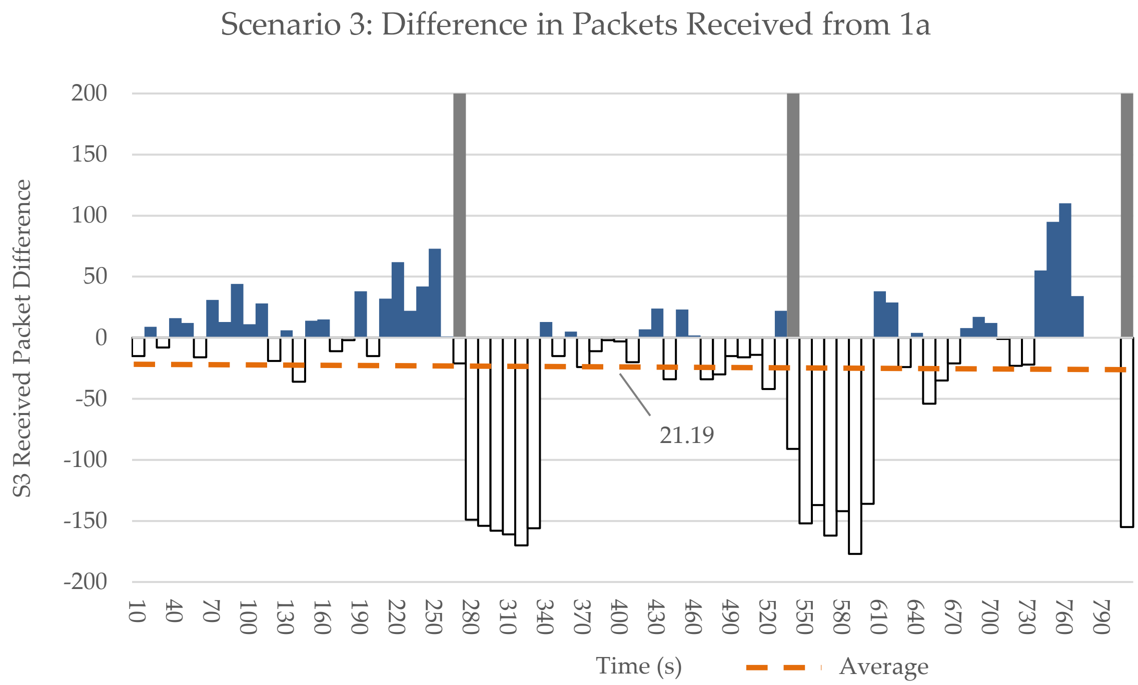


Figure 50. The difference in received packets in scenario 3 as compared with scenario 1a. Restricted ground access results in clear reductions in throughput during the opening periods of the second and third passes.

Scenario 3 does not experience scenario 1a's anomaly as might be suggested by packet reception results during the third ground pass (Figure 49). The observed drop in packet reception occurs almost 30s later than the drop caused by the scenario 1a anomaly. In scenario 3, The GM role is explicitly removed from nodeMaster[2] during this period which is not the case in scenario 1a.

D^3's energy-distance election approach results in nodeMaster[0] being rested during the opening periods of the second and third passes due to its elevated energy consumption during the first pass (Figure 49, Figure 50). NodeMaster[2] is not rested in a similar manner during the second pass. This does not represent unexpected behaviour. The energy-distance election approach stretches GM durations of the 'inner' masters during the first pass. This effectively reduces the load on nodeMaster[2]. The first pass beings with all nodes having consumed zero Joules of energy. As such, it is not possible to relieve the pressure on nodeMaster[0] in a similar manner to nodeMaster[2].

Overall, scenario 3's ground station receives 1716 fewer packets than scenario 1a's ground station and 2796 fewer than scenario 2a's ground station. Within each 10 second result interval scenario 3 achieves similar packet reception rates. However, the "resting" of nodeMaster[0] and nodeMaster[2] reduces the overall S2G communication time which is available to the CSN. As these masters are the leading and trailing masters of the CSN respectively, there are periods during which these masters are the only masters within communication range of ground. In comparison, nodeMaster[1] could be completely removed from the CSN and the total time available for S2G communication would remain unchanged.

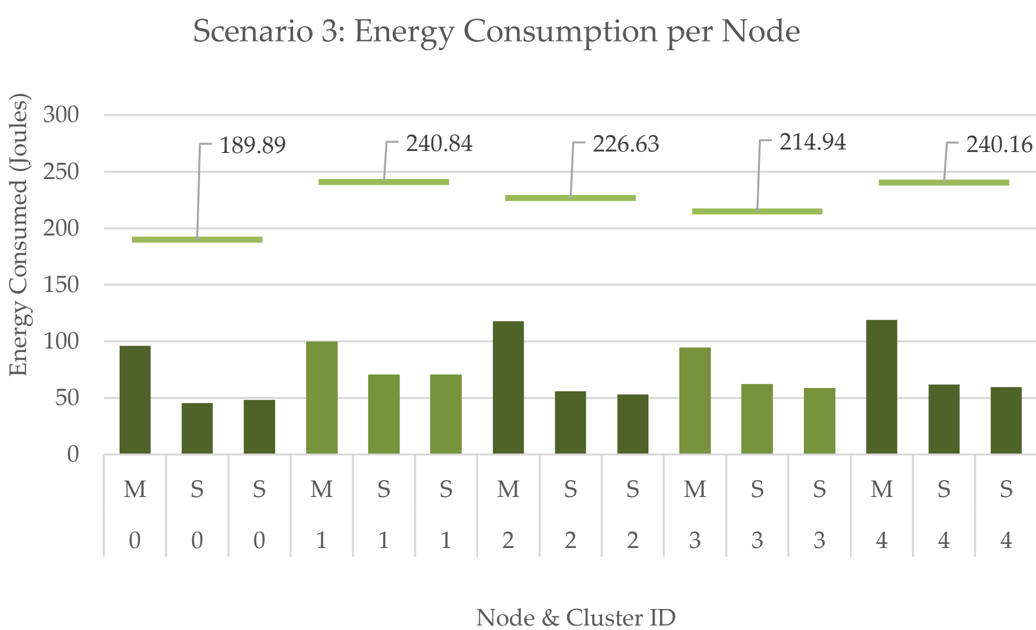


Figure 51. Due to scenario 3's altered election approach masters experience an overall drop in energy consumption when compared with scenario 1a and 2a.

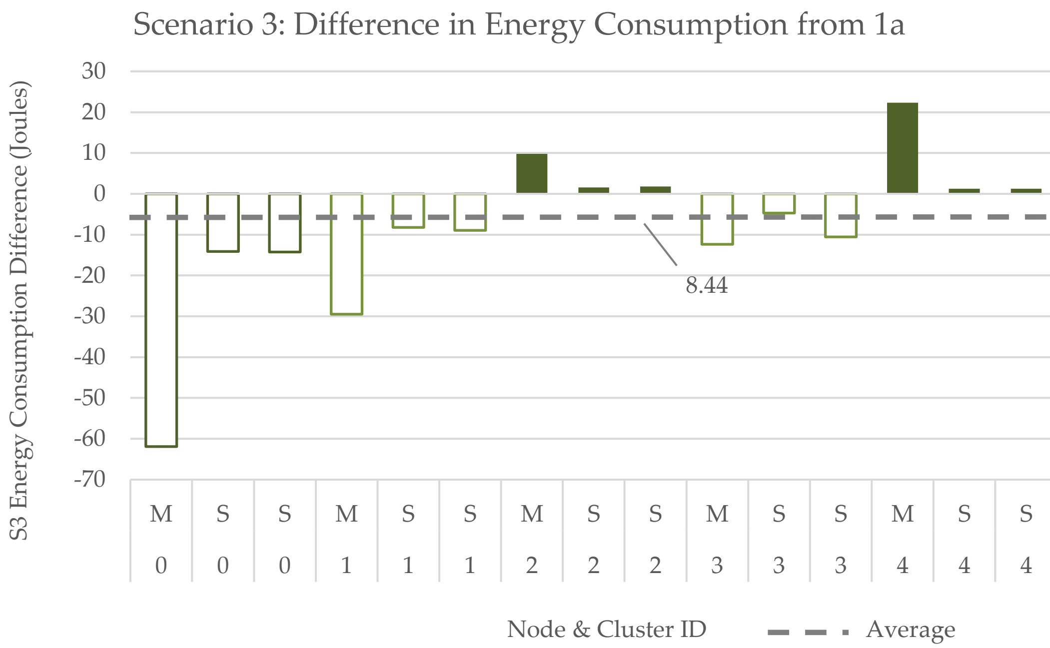


Figure 52. Negative (outlined) columns represent the energy saved by individual nodes as a result of D³'s energy-distance election approach.

The standard deviation between master energy consumption totals in scenario 1a was ~30J. In scenario 2a this standard deviation is ~32J. In comparison, this standard deviation in scenario 3 is ~24J. This result is illustrated in the reduction of the differences between master energy consumption totals shown in Figure 51 and Figure 42. The reduction in standard deviation represents the success of D³'s energy-distance GM election approach in balancing the GM workload more evenly across the CSN's masters. The closest-master approach fixes GM durations for each master (Table 3). In contrast, the energy-distance approach allows for flexible GM durations which relate directly to a master's energy consumption. The energy-distance election approach favours masters that have consumed less energy relative to the CSN's other masters. This increased favourability results in an increase in the GM duration of masters which have consumed less energy overall (Figure 53). Similarly, masters which have consumed more energy overall will have their GM durations reduced. The energy-distance approach is expected to cause all GM durations to converge on approximately the same value, given consistent traffic patterns.

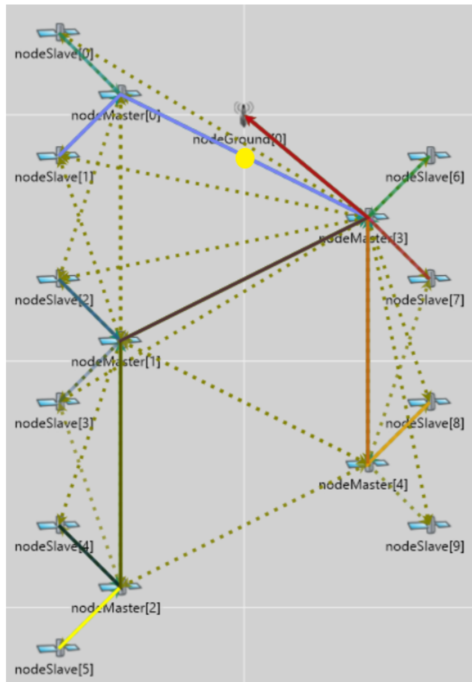


Figure 53. The "early" election of nodeMaster[3] using the energy-distance election approach. The point at which nodeMaster[3] would be elected GM using the closest-master approach is represented by a yellow circle.

5.6 Efficiency

The average amount of energy required to communicate one packet to ground is a key performance indicator for the PvTP trade-off. Table 4 shows that all scenarios apart from scenario 2a are less efficient than scenario 1a. In the case of scenario 1b this is expected as CubeMac's additional energy saving features are not utilized. Scenario 2a's improved efficiency may be reflective of a more efficient approach, however, scenario 1a's anomaly calls this result into question. Scenario 2b is, as expected, considerably less efficient. Finally, scenario 3's drop in efficiency is unexpected. Scenario 1a's anomaly tends to falsely improve the performance of scenarios compared with scenario 1a. Despite this boost from scenario 1'a anomaly, scenario 3 still shows a lower S2G packet energy efficiency than scenario 1a. This suggests that scenario 3 is less efficient in comparison to scenario 1a than suggested by the results in Table 4.

The most likely source of scenario 3's reduced efficiency is the energy consumed by route discovery attempts. Scenario 3 and scenario 1a both involve a similar number of GM role transferrals. Each change in GM causes all node nodes to drop their routes to ground and start a route discovery cycle. Each cycle incurs a fixed energy overhead. In scenario 3 the total number of packets received reduces which increases the overall proportion of route discovery energy consumption to packets received. In short, the impact of the energy-distance approach on throughput is not matched by a proportional decrease in energy consumed by route discovery.

Table 4. A summary of metric totals presented alongside a key performance indicator; the approximate amount of energy required to send a single packet to ground. Change figures represented in green indicate positive change, i.e. an improvement, over scenario 1a and vice versa.

Scenario	Packets	Change	Energy (J)	Change	Energy/Packet	Change
1a	10771	0%	1239.099	0%	0.11504	0%
1b	10771	0.00%	1271.686	2.63%	0.118066	2.63%
2a	11851	10.03%	1287.153	3.88%	0.108611	-5.59%
2b	771	-92.84%	667.1736	-46.16%	0.865335	652.20%
3	9055	-15.93%	1112.475	-10.22%	0.122858	6.80%

Chapter 6: Conclusions

This work presents an exploration of the CSN PvTP trade-off through the examination of several areas of relevant research and the development and analysis of two potential CSN communication protocols. Despite the reduction of the overall scope of this work to a generalized hypothetical mission, considerable background information is required to adequately assess and approach the CSN PvTP trade-off.

This work introduces a general background which details the relevant state-of-the-art of the CubeSat platform and CubeSat missions. Relevant CubeSat capabilities and CubeSat applications are explored in depth. This exploration provides important context and illustrates several of the salient realities of CubeSat missions. Three major areas of relevant prior art are investigated; WSNs, MANETs, and CubeSat communications. Works relating to WSNs were found to be more relevant than those relating to MANETs due to a greater treatment of resource constraints and node failures. Works relating to CubeSat communications provided a strong basis for this work's proposed MAC protocol. However, a notable lack of relevant routing protocol related work was identified. In place of works related to CubeSat routing, MANET related works informed the development of this work's proposed routing protocol.

This work's proposed protocols are intended to address the CSN PvTP trade-off. Further work is required to increase the fidelity of CSN simulation and the assessment of proposed protocols. Despite this, the assessment of protocol modifications showed the potential benefits of CubeMac's energy saving features and D^3's energy-distance GM election. Such modifications affect the PvTP trade-off and represent tangible contributions to the field of CubeSat communications.

6.1 Discussion

This section provides discussion on the validity and accuracy of simulation results and the potential value of this work's contributions. Where possible, the impact of this work's findings is presented in the wider context of CubeSats and the space industry. This section concludes with discussions of several broad topics and space sector trends relating to CSNs.

6.1.1 Results

The anomaly observed in scenario 1a damages the validity of this work's results. This anomaly is especially damaging in its effect on the assessment of CubeMac's pure TDMA mode in scenario 2a. Certain findings are less impacted by scenario 1a's anomaly. For instance, the poor performance of the INET CSMA protocol in scenario 2b is expected to a degree due to it its contention based nature and the prior work of Radhakrishnan et al. Also, scenario 1b's degraded energy illustrates the value of CubeMac's added energy saving features despite scenario 1a's anomaly. The validity of Scenario 3's results were largely unaffected by the scenario 1a anomaly. Scenarios 3's energy-distance approach was shown to be less energy efficient overall than the closest-master approach despite a boost in apparent efficiency due to scenario 1a's anomaly.

Notable simplifications and assumptions made by this work's base simulation are detailed in chapter 4. It worth reiterating that the base simulation reflects only a small subset of the expected properties of CSNs. Also, considering the quantity and severities of issues encountered during development, it is possible that this work's results are fundamentally skewed by unidentified issues within OMNeT++ and/or the INET framework. The fact that the simulation resources utilized by this work were not explicitly designed for the simulation of satellite communications should be considered when assessing the fidelity of the results presented by this work.

An assumption made regarding CDMA based communication in the base simulation has a considerable impact on results relating to the difference in energy consumption between masters and slaves. It is assumed that CDMA based communication requires 16% more energy. It was not possible to identify a relevant reference for the increase in energy consumption due to CDMA. It is likely that this assumption is conservative and assumes a very basic CDMA scheme. Also, several unaddressed aspects of CDMA may affect energy consumption. Further work is required to establish a well-founded value for CDMA's additional energy cost.

Section 2.1.1 details the battery and recharge capabilities of the EDSN mission. The average energy consumed by masters in scenario 1a was ~120J. A rough estimate places this energy consumption at $7.24x10^{-4}$% of an EDSN craft's total available energy. This percentage is divided by three to compute an average energy consumption per pass of $2.41x10^{-4}$%. Over the total duration of the EDSN mission this would result in the consumption of ~11% of a crafts available energy. Although, a CubeSat's radios typically only consume a small proportion of the overall energy budget, the validity of this result is dubious. This result may be invalidated due to assumptions made in this work's simulation of communication power requirements. The probable inaccuracy of this work's energy consumption result values is not considered to invalidate their use for the comparison of simulation scenarios.

6.1.2 Contributions

This work provides an exploration and assessment of various notable relevant areas of research and technology which provide background to the CSN PvTP trade-off. Prior art related to CubeSats and CubeSat communications is covered in greater depth than the terrestrial fields of WSN's and MANETs. This work identifies broad trends in these fields and assesses their relevancy to CSNs. Further work is required in order to identify and assess specific WSN and

MANET approaches and technologies for application within the field of CSNs. Nonetheless, this work provides a guide to the examination of relevant terrestrial communications and contributes in its broad assessments thereof.

This work also contributes in its general approach to the assessment of the CSN PvTP trade-off. The design of a simplified hypothetical mission informs the development of simulation scenarios. These scenarios are developed through the use of open-source network simulation resources. The various subjective strengths and weaknesses of this work's approach may be used to inform the approach taken by future work in the exploration CSN related topics.

CubeMac

This work's proposed MAC protocol contributes to the field of CubeSat communications in two regards. CubeMac recreates major aspects of the work of Radhakrishnan et al. within OMNeT++'s INET framework. This recreation may benefit future developments or assessments of Radhakrishnan et al.'s underlying C/TDMA protocol. Alongside several modifications to C/TDMA, CubeMac introduces two novel energy saving features. These features were shown, through simulation, to reduce energy consumption without negatively affecting S2G throughput.

Several aspects of the implementation of CubeMac reduce the fidelity of its recreation of C/TDMA. In particular, simulated CDMA based communication is highly simplified which may impact the reusability of CubeMac's OMNeT++ implementation. Also, CubeMac's deviation from the "uplink – downlink" frame structures of C/TDMA reduces its value as a recreation of Radhakrishnan et al.'s work.

CubeMac lacks features enabling the dynamic formation of clusters and election cluster masters. These features, as discussed in section 3.1, are critical in real world applications of CubeMac.

Results presented in section 5.3 illustrate that omitting CubeMac's cluster architecture reduces the variation in node energy consumption totals and causes a rise in packet E2E delays. Clustering could not be stated to have a significant impact on S2G throughput. The additional overheads of cluster formation and maintenance may impact the approach's favourability over non-cluster based approaches. C/TDMA's cluster architecture's ability to reduce the energy consumption of slave nodes appears to constitute the approach's primary benefit in relation to the PvTP trade-off.

D^3

D^3 related contributions centre on the addition of the ground master role. The majority of the development resources committed to D^3 were dedicated to resolving fundamental issues with INET's DYMO module. It is likely that D^3's OMNeT++ module retains as yet unidentified bugs and errata. Nonetheless, the module may contribute to future, more complete, implementations of IETF DYMO specifications for the INET framework.

D^3's use of an oracle to perform GM election and simulate RERRs is undesirable but proved necessary given this work's limited development resources. Aside from D^3's oracle, this work avoids the treatment of several aspects of the GM role and the intermittent nature of S2G communications. For instance, only one active GM is permitted by D^3. Each S2G link represents a valuable, yet costly, resource which provides the basis for the unique nature of CSNs. As such, D^3's performance within a CSN may be assessed on its use of available S2G links.

This work assumes that S2G communication occurs on a frequency shared by all CubeSats. However, if multiple CubeSats could communicate to single ground station in parallel, using a CDMA MAC protocol for example, then the introduction of multiple GMs may greatly increase throughput and reduce wasteful S2S communication. Without parallel S2G communication, CSNs may still benefit from the introduction of multiple concurrent GMs. For instance, a master may assume the GM role prior to its S2G communication window and announce a "virtual" route to ground. Through D^3's use of link costs, nearby nodes may begin to forward their packets to this "virtual" GM rather than the current "active" GM. This approach could alleviate the negative effects of a GM receiving more packets than can be communicated to ground during duration as GM.

D^3's design requires masters which are exiting the GM role to forward surplus packets onto the next GM. This has the potential to produce wasteful S2S communication. For instance, if an exiting master knows it will obtain the GM role again in future it may be beneficial for this master to temporarily avoid forwarding its queued packets. This master may decide to begin forwarding packets again as it approaches the maximum number of packets which may be reasonably communicated to ground. In general, D^3 underutilizes both the semi-predictable order of GM elections and the predicable durations of S2G communication windows.

GM election may be modified to account for the current number of packets queued by masters. Election may also consider the proportion of these queued packets which can be communicated to ground during an upcoming S2G communication window. An election approach based on these factors may include a feature allowing the removal of all GMs from a CSN. This feature could allow nodes to hold-off on S2S communication until a master with suitable resources is elected as GM.

A similar hold-off feature could also be implemented by allowing GMs to generate special "hold-off" packets. Such packets, when broadcast throughout the network, could inform nodes to stop sending packets to an overloaded GM. Also, a hold-off packet may specify that nodes should not attempt any further route discovery for a given period in order to avoid wasteful route message traffic. This highlights another fundamental flaw of D^3. In simulation scenario 3, D^3 deliberately produces periods wherein no GM is elected. During these periods, there is no available route to ground. However, nodes continue to attempt route discovery. This further illustrates the potential need for a mechanism which may signal nodes to reduce the frequency of route discovery attempts.

D^3 only scratches the surface of CSN related routing protocol challenges. The D^3 energy-distance GM election approach represents this work's small contribution to the CSN routing protocol domain.

Simulation

OMNeT++ and INET framework related issues consumed the majority of the development time available to this work. It is estimated that approximately 75% of the time spent developing this work's proposed protocols and simulation scenarios was dedicated to addressing unexpected issues. The notable issues encountered during development are discussed in section 4.2. The identification and resolution of these issues may allow future researchers to dedicate a larger proportion of their development resources to the advancement and assessment of communication protocols and CSN simulations.

6.1.3 Space Junk

The rise of in-orbit space junk, or space debris [127], has caused growing concerns within the space sector over the past two decades. No formal works were identified which address the potential impact of CSNs on the growth of LEO space junk. However, several works address the impact of CubeSat missions on the growth of space junk. A single CubeSat may remain in orbit for several years beyond its operational lifetime or mission duration [128]. At present, there are no universal requirements placed upon CubeSat mission designers to provide mechanisms to de-orbit CubeSats in a timely manner. Considering the growth in the number of CubeSat missions over the past decade, CubeSat's stand to contribute significantly to the ongoing growth of space junk.

Technologies capable of safely de-orbiting CubeSats have been developed by various groups in response to growing space junk concerns [129]. If international regulations change, such technologies may become mandatory in future CubeSat missions. International regulation aimed at the reduction of space junk may pose a threat to the growth of CSN missions. For instance, regulation introduced to slow the growth of space junk may place limits on the number of CubeSats which can be launched into a single formation. In any case, there is clear need for widespread and reliable CubeSat de-orbiting technologies in future missions. In general, space junk restricts access to LEO by reducing the number of available orbits and can cause fatal damage to orbiting spacecraft. It is to the benefit of all those in space sector to slow the growth of space junk and, where possible, reduce existing space junk.

6.1.4 Mission Design

Once a satellite is launched, mission operators have limited power to resolve issues. As such, space-bound technologies are thoroughly tested and simplified. There are hints of this in the design of NASA's EDSN and Nodes missions. Mission designers opted to design a protocol rather than implement an existing one. This has two potential benefits; it allows for complete knowledge of communication behaviour by mission operators and reduces the risk of the existence of unknown bugs or flaws. Both CubeMac and D^3 are less complex than certain state-of-the-art MAC and routing protocols. However, less complex protocols exist. For instance, static TDMA and DSR. Although such protocols may be less performant, their reduced complexity may make them favourable in future CSN missions.

6.1.5 Remote and Extreme Environments

Future CSNs may provide low bandwidth communications for remote and extreme environments. CubeSat S2G downlink speeds are unlikely to reach levels suitable to provide consistent internet services within the next decade. However, provided a suitably capable S2G uplink is available, orbiting CSNs may relay basic communications data from ground via S2S links to more capable craft. For instance, uplinks performed from regions affected by natural disasters could provide vital information to incoming emergency services.

Due to their low cost, CSNs may one day provide comprehensive coverage of Earth's surface through the formation of constellations. In a similar manner to certain modern GNSS constellations, CSN constellations may form as a result of the inter-communication of multiple CSN missions. With comprehensive cover of earth's surface CSNs may provide an extension to low bandwidth communication services at a price point far below that of existing satellite infrastructure.

6.1.6 CubeSats Beyond LEO

NASA's Space Launch System, which is currently estimated to make its debut flight in 2019, will open access to deep space for CubeSats (Figure 5). Alongside SLS, other "heavy lift" launch platforms such as SpaceX's upcoming "Flacon Heavy" and Blue Origin's "New Glenn" will further extend the reach of CubeSats beyond LEO. Several CubeSat researchers have begun work on the developments in CubeSats capabilities required to take the platform beyond LEO. For instance, NASA's JPL has produced work on a CubeSat design referred to as "Lunar Flashlight" [130].

ESA's Asteroid Impact Mission (AIM) mission proposed a potential first deep space application of a CSN [131]. The CubeSat Opportunity Payloads (COPINs) component of the AIM mission involved the deployment of up to three CubeSats alongside AIM's primary satellite and asteroid lander. These CubeSats were primarily intended as a technology demonstration of CubeSat operations in deep space. The CubeSat's objectives were to perform measurements of particulate matter surrounding AIM's target asteroids. The CubeSat's were to be tasked with collaboratively communicating their data to the primary AIM craft. Although the AIM mission was defunded, AIM-COPINs is a clear indication of the potential value of CubeSats beyond LEO.

CSN's open numerous opportunities for the novel observation of extra-terrestrial bodies. For instance, consider NASA's well-known voyager missions. Collectively, voyager craft performed fly-bys of effectively all major celestial bodies within the solar system. Such future fly-bys and gravity assists present an opportunity for CSNs. CubeSat's deployed during a fly-by may perform novel autonomous observations of celestial bodies. Through the establishment of a formation flying CSN, the CubeSats may opportunistically relay their data to a parent craft or other more capable craft for relay to Earth.

CSNs are in their infancy in LEO. However, without the availability of nearby craft or multiple ground stations, the formation of CSNs may become a necessity for CubeSats beyond LEO. CSNs increase mission complexity but reduce the impact of failures and allow for greater utilization of limited CubeSat resources.

6.2 Future Work

The section presents several opportunities for future work based on the contributions and findings of this work. Several unaddressed research problems are noted as areas for future research efforts. Several proposed areas of future work have been discussed in the preceding sections. As such, this section provides listings of areas of future work rather than detailed discussions thereof.

6.2.1 Data Dissemination

This work focuses on data collection, as discussed in relation to WSNs. NASA's Nodes investigates, to an extent, the dissemination of command and control data across a single hop CSN. D^3 illustrates basic dissemination of route messages throughout a CSN.

Proposed areas of future work:

- Unscheduled management and control of individual CubeSat and CSN behaviour from ground in multi-hop CSNs
- The distribution of software or firmware updates across members of a CSN
- The efficacy and reliability of sharing individual CubeSat GNSS and status information to form CSN topology models and estimate ranging

6.2.2 Expanded CSNs

The CSN simulated in this work consists of fifteen CubeSats in LEO. This represents a considerable advancement from TW-1's state-of-the-art three node CSN. Several proposed CSN missions implement larger and more complex networks.

Proposed areas of future work:

- Large CSNs composed of multiple CubeSat swarms in unsynchronized disparate orbits
- The feasibility and requirements of a CSN constellations providing varying degrees of coverage of Earth's surface

- The opportunistic use of larger more capable craft to relay data to and from ground

6.2.3 CubeMac

Proposed areas of future work:

- Dynamic clustering and master election in response to factors such as: node mobility, S2G communication windows and/or node power capabilities
- Examination of the increased cost of performing basic CDMA within TDMA time slot schedules
- Acknowledgement and error correction functionality
- Local assignment and adaptive sharing of time slots
- Asynchronous adjustment of slot durations in response to traffic requirements
- A comparative performance study with LDMA and LMAC

6.2.4 D^3

Proposed areas of future work:

- Replacement of the stand-in oracle approach with a distributed decision making approach and correct RERR message handling
- GM election considering all CSNs nodes rather than a dedicated subset thereof
- Assignment of 'active' and 'virtual' GMs to reduce packet handover overheads
- GM hold-off through early GM removal or dedicated hold-off requests
- Decision making performed by to-be GMs to determine the optimal number of packets to hold on S2G queues
- Implicit election of GMs through adaptive link cost functions

6.2.5 CSN Simulation

Proposed areas of future work:

- Replacement of 'Ideal' INET modules with modules parameterized to reflect realistic CubeSat communications and signal propagation in LEO
- Evaluation of IPv6 related technologies for use with large CSNs such as RPL
- Battery performance, energy consumption and energy generation models based upon the known performance of tested CubeSat components
- Simulation of the properties of the CSN applications proposed by Challa and McNair of the University of Florida [99-102]
- Accurate OMNeT++ mobility models and simulation durations of realistic three dimensional orbits with the potential dynamic effects of orbital perturbations

This section gets auto formatted by EndNote on updates, it will be the last section to get edited and properly formatted.

References

[1] L. Brennan and A. Vecchi, *The business of space: The next frontier of international competition*: Palgrave Macmillan, 2011.

[2] G. Johnson. (2012, January, 15). *Revised, Expanded Launch Cost Data*. Available: http://exrocketman.blogspot.ie/2012/05/revised-expanded-launch-cost-data.html

[3] S. Korus. (2016, May 15th). *Reusable Rockets: At the First Stage*. Available: https://ark-invest.com/research/reusable-rockets

[4] A. Scholz and J.-N. Juang, "Toward open source CubeSat design," *Acta Astronautica,* vol. 115, pp. 384-392, 2015.

[5] S. Padmanabhan, S. Brown, B. Lim, P. Kangaslahti, D. Russell, and R. Stachnik, "Airborne Deployment and Calibration of Microwave Atmospheric Sounder on 6U CubeSat," in *AGU Fall Meeting Abstracts*, 2015.

[6] V. Hernandez, P. Gankidi, A. Chandra, A. Miller, P. Scowen, H. Barnaby*, et al.*, "SWIMSat: Space Weather and Meteor Impact Monitoring using a Low-Cost 6U CubeSat," 2016.

[7] U. Kvell, M. Puusepp, F. Kaminski, J.-E. Past, K. Palmer, T.-A. Grönland*, et al.*, "Nanosatellite orbit control using MEMS cold gas thrusters," *Proceedings of the Estonian Academy of Sciences,* vol. 63, p. 279, 2014.

[8] X. Sun and X. Wu, "A cubesat attitude control system with linear piezoelectric actuator," in *Piezoelectricity, Acoustic Waves, and Device Applications (SPAWDA), 2014 Symposium on*, 2014, pp. 72-75.

[9] A. Budianu, T. J. W. Castro, A. Meijerink, and M. J. Bentum, "Inter-satellite links for cubesats," in *Aerospace Conference, 2013 IEEE*, 2013, pp. 1-10.

[10] R. Radhakrishnan, W. W. Edmonson, F. Afghah, R. M. Rodriguez-Osorio, F. Pinto, and S. C. Burleigh, "Survey of Inter-satellite Communication for Small Satellite Systems: Physical Layer to Network Layer View," *IEEE Communications Surveys & Tutorials,* vol. 18, pp. 2442-2473, 2016.

[11] Y. F. Wong, O. Kegege, S. H. Schaire, G. Bussey, S. Altunc, Y. Zhang*, et al.*, "An Optimum Space-to-Ground Communication Concept for CubeSat Platform Utilizing NASA Space Network and Near Earth Network," 2016.

[12] W. Harrington and J. Heath, "Development of a Low-Cost, Open Software/Hardware Command, Control and Communications Module for CubeSats," in *AIAA SPACE 2016*, ed, 2016, p. 5616.

[13] J. Hanson, A. G. Luna, R. DeRosee, K. Oyadomari, J. Wolfe, W. Attai*, et al.*, "Nodes: A Flight Demonstration of Networked Spacecraft Command and Control," 2016.

[14] A. Tatomirescu, G. F. Pedersen, J. Christiansen, and D. Gerhardt, "Antenna system for nano-satelite mission GOMX-3," in *Antennas and Propagation in Wireless Communications (APWC), 2016 IEEE-APS Topical Conference on*, 2016, pp. 282-285.

[15] M. Swartwout, "The first one hundred CubeSats: A statistical look," *Journal of Small Satellites,* vol. 2, pp. 213-233, 2013.

[16] K. Kelley, "Launch systems to support the booming nanosatellite industry," in *Aerospace Conference, 2015 IEEE*, 2015, pp. 1-6.

[17] D. Hitt, K. F. Robinson, and S. D. Creech, "NASA's Space Launch System: A New Opportunity for CubeSats," 2016.

[18] D. Masutti, T. Banyai, J. Thoemel, T. Magin, B. Taylor, and D. Kataria, "Investigating the Middle and Lower Thermosphere using a Cubesat Constellation: the QB50 Mission and its Particular Challenges," in *EGU General Assembly Conference Abstracts*, 2015, p. 9016.

[19] M. Tsay, J. Frongillo, K. Hohman, and B. K. Malphrus, "LunarCube: A Deep Space 6U CubeSat with Mission Enabling Ion Propulsion Technology," 2015.

[20] R. W. Ridenoure, D. A. Spencer, D. A. Stetson, B. Betts, R. Munakata, S. D. Wong, *et al.*, "Status of the Dual CubeSat LightSail Program," in *AIAA SPACE 2015 Conference and Exposition*, 2015, p. 4424.

[21] R. Glumb, C. Lietzke, S. Luce, and P. Wloszek, "Cubesat Fourier Transform Spectrometer (CubeSat-FTS) for Three-Dimensional Global Wind Measurements," in *American Meteorological Society Annual Meeting,(January 2015)*, 2015.

[22] S. Nag, J. L. Rios, D. Gerhardt, and C. Pham, "CubeSat constellation design for air traffic monitoring," *Acta Astronautica,* vol. 128, pp. 180-193, 2016.

[23] D. Westley, A. Martinez, and A. Petro, "Edison Demonstration of Smallsat Networks," 2015.

[24] R. Barbosa. (2015, September, 24). *China debuts Long March 11 lofting Tianwang-1 trio*. Available: NASASpaceFlight.com

[25] A. Varga, "OMNeT++," *Modeling and tools for network simulation,* pp. 35-59, 2010.

[26] G. Bora, S. Bora, S. Singh, and S. M. Arsalan, "OSI reference model: An overview," *International Journal of Computer Trends and Technology (IJCTT,* vol. 7, 2014.

[27] H. Helvajian and S. W. Janson, *Small satellites: past, present, and future*: Aerospace Press, 2008.

[28] H. Heidt, J. Puig-Suari, A. Moore, S. Nakasuka, and R. Twiggs, "CubeSat: A new generation of picosatellite for education and industry low-cost space experimentation," 2000.

[29] M. Swartwout, "Cubesat database," *St. Louis University.[Online].[Accessed 7 February 2015],* 2015.

[30] C. Boshuizen, J. Mason, P. Klupar, and S. Spanhake, "Results from the planet labs flock constellation," 2014.

[31] R. A. Deepak and R. J. Twiggs, "Thinking out of the box: Space science beyond the CubeSat," *Journal of Small Satellites,* vol. 1, pp. 3-7, 2012.
[32] J. Puig-Suari, J. Schoos, C. Turner, T. Wagner, R. Connolly, and R. Block, "CubeSat developments at Cal Poly: the standard deployer and PolySat," in *Proceedings of SPIE-The International Society for Optical Engineering*, 2000, pp. 72-78.
[33] J. Farkas, "CPX: Design of a standard cubesat software bus," *California State University, California, USA,* 2005.
[34] J. Straub, C. Korvald, A. Nervold, A. Mohammad, N. Root, N. Long, *et al.*, "OpenOrbiter: A low-cost, educational prototype CubeSat mission architecture," *Machines,* vol. 1, p. 1, 2013.
[35] A. K. Nervold, J. Berk, J. Straub, and D. Whalen, "A Pathway to Small Satellite Market Growth," *Advances in Aerospace Science and Technology,* vol. 1, p. 14, 2016.
[36] W. A. Beech, D. E. Nielsen, J. T. Noo, and L. K. Ncuu, "AX. 25 Link Access Protocol for Amateur Packet Radio, Version: 2.2 Rev," in *Tucson Amateur Packet Radio Corp*, 1997.
[37] P. Muri and J. McNair, "A survey of communication sub-systems for intersatellite linked systems and CubeSat missions," *JCM,* vol. 7, pp. 290-308, 2012.
[38] C. Fish, C. Swenson, T. Neilsen, B. Bingham, J. Gunther, E. Stromberg, *et al.*, "Dice mission design, development, and implementation: Success and challenges," 2012.
[39] R. Hodges, B. Shah, D. Muthulingham, and T. Freeman, "ISARA–Integrated Solar Array and Reflectarray Mission Overview," 2013.
[40] S. Wu, W. Chen, and C. Chao, "The STU-2 CubeSat Mission and In-Orbit Test Results," 2016.
[41] S. S. Arnold, R. Nuzzaci, and A. Gordon-Ross, "Energy budgeting for CubeSats with an integrated FPGA," in *Aerospace Conference, 2012 IEEE*, 2012, pp. 1-14.

[42] A. Oliveira. (2015, 02-02). *Final Report Summary - GAMALINK (Generic SDR-bAsed Multifunctional spAce LINK)*. Available: http://cordis.europa.eu/result/rcn/172006_en.html

[43] (2016, April 14th). *CubeSat Design Overview Report*. Available: http://sydney.edu.au/engineering/aeromech/AERO3760/private/CDR/1%20%20Critical%20Design%20Overview%20i-INSPIRE%EF%BC%92.pdf

[44] J. Guo, J. Bouwmeester, and E. Gill, "From Single to Formation Flying CubeSats: An Update from the Delft Programme," 2013.

[45] M. Focardi, V. Noce, S. Buckley, K. O'Neill, A. Bemporad, S. Fineschi, *et al.*, "The shadow position sensors (SPS) formation flying metrology subsystem for the ESA PROBA-3 mission: present status and future developments," in *SPIE Astronomical Telescopes+ Instrumentation*, 2016, pp. 99044Z-99044Z-17.

[46] J. M. EncinasPlaza, J. A. VilanVilan, F. AquadoAgelet, J. BrandiaranMancheno, M. LopezEstevez, C. MartinezFernandez, *et al.*, "Xatcobeo: Small Mechanisms for CubeSat Satellites-Antenna and Solar Array Deployment," 2010.

[47] G. Sun, X. Xia, S. Wu, Z. Wu, and W. Chen, "Attitude Determination and Control System Design for STU-2A CubeSat and In-Orbit Results," 2016.

[48] J. Li, M. Post, T. Wright, and R. Lee, "Design of attitude control systems for cubesat-class nanosatellite," *Journal of Control Science and Engineering,* vol. 2013, p. 4, 2013.

[49] E. Glennon, J. Gauthier, M. Choudhury, A. Dempster, and K. Parkinson, "Synchronization and syntonization of formation flying cubesats using the namuru V3. 2 spaceborne GPS receiver," in *Proceedings of the the ION 2013 Pacific PNT Meeting, Honolulu, HI, USA*, 2013, pp. 23-25.

[50] A. Cortiella, D. Vidal, J. Jané, E. Juan, R. Olivé, A. Amézaga, *et al.*, "3CAT-2: Attitude Determination and Control System for a GNSS-R Earth Observation 6U CubeSat Mission," *European Journal of Remote Sensing,* vol. 49, pp. 759-776, 2016.

[51] W. H. Swartz, S. R. Lorentz, P. M. Huang, A. W. Smith, D. M. Deglau, S. X. Liang*, et al.*, "The Radiometer Assessment using Vertically Aligned Nanotubes (RAVAN) CubeSat Mission: A Pathfinder for a New Measurement of Earth's Radiation Budget," 2016.

[52] W. H. Swartz, L. P. Dyrud, S. R. Lorentz, D. L. Wu, W. J. Wiscombe, S. J. Papadakis*, et al.*, "The RAVAN CubeSat mission: advancing technologies for climate observation," in *Geoscience and Remote Sensing Symposium (IGARSS), 2015 IEEE International*, 2015, pp. 5300-5303.

[53] S. Kanekal, P. O'Brien, D. N. Baker, K. Ogasawara, J. Fennell, E. Christian*, et al.*, "Radition belt dynamics: Recent results from van Allen Probes and future observations from CeREs," in *41st COSPAR Scientific Assembly, abstracts from the meeting that was to be held 30 July-7 August at the Istanbul Congress Center (ICC), Turkey, but was cancelled. See* http://cospar2016*. tubitak. gov. tr/en/, Abstract PRBEM. 2-1-16.*, 2016.

[54] J. Westerhoff, G. Earle, R. Bishop, G. R. Swenson, S. Vadas, J. Clemmons*, et al.*, "LAICE CubeSat mission for gravity wave studies," *Advances in Space Research,* vol. 56, pp. 1413-1427, 2015.

[55] J. M. Morrison, H. Jeffrey, H. Gorter, P. Anderson, C. Clark, A. Holmes*, et al.*, "SeaHawk: an advanced CubeSat mission for sustained ocean colour monitoring," in *SPIE Remote Sensing*, 2016, pp. 100001C-100001C-11.

[56] M. Bentum, A. Meijerink, A.-J. Boonstra, C. Verhoeven, and A.-J. v. d. Veen, "OLFAR: the orbiting low frequency array, how a cube sat swarm becomes a novel radio astronomy instrument in space," *De Vonk,* vol. 25, pp. 1-5, 2010.

[57] J. Hanson, J. Chartres, H. Sanchez, and K. Oyadomari, "The EDSN intersatellite communications architecture," 2014.

[58] J. Chartres, H. Sanchez, and J. Hanson, "EDSN development lessons learned," 2014.

[59] K. I. Parker, "State-of-the-Art for Small Satellite Propulsion Systems," 2016.

[60] O. Barnouin, J. Biele, I. Carnelli, V. Ciarletti, A. Cheng, A. Galvez, *et al.*, "The Asteroid Impact and Deflection Assessment (AIDA) mission: Science Proximity Operations," in *LPSC 2016 47th Lunar and Planetary Science Conference*, 2016, p. 1427.
[61] M. Bisgaard, D. Gerhardt, H. Hermanns, J. Krčál, G. Nies, and M. Stenger, "Battery-Aware Scheduling in Low Orbit: The GomX–3 Case," in *FM 2016: Formal Methods: 21st International Symposium, Limassol, Cyprus, November 9-11, 2016, Proceedings 21*, 2016, pp. 559-576.
[62] B. Niels, "ESA and GomSpace Sign Contract to Launch Advanced Nanosatellite," ed. Web: GOMspace, 2016.
[63] M. Villa, A. Martinez, and A. Petro, "Cubesat Proximity Operations Demonstration (CPOD)," 2015.
[64] D. Cecil, "Potential Future NASA Satellite Data and Applications for Tropical Cyclones," 2016.
[65] E. Gill, P. Sundaramoorthy, J. Bouwmeester, B. Zandbergen, and R. Reinhard, "Formation flying within a constellation of nano-satellites: The QB50 mission," *Acta Astronautica,* vol. 82, pp. 110-117, 2013.
[66] T. Rault, A. Bouabdallah, and Y. Challal, "Energy efficiency in wireless sensor networks: A top-down survey," *Computer Networks,* vol. 67, pp. 104-122, 2014.
[67] J. Zheng and M. J. Lee, "A comprehensive performance study of IEEE 802.15. 4," ed: IEEE Press book Los Alamitos, 2004.
[68] C. Gomez, J. Oller, and J. Paradells, "Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology," *Sensors,* vol. 12, pp. 11734-11753, 2012.
[69] N. Accettura, L. A. Grieco, G. Boggia, and P. Camarda, "Performance analysis of the RPL routing protocol," in *Mechatronics (ICM), 2011 IEEE International Conference on*, 2011, pp. 767-772.
[70] P. T. A. Quang and D.-S. Kim, "Throughput-aware routing for industrial sensor networks: Application to ISA100. 11a," *IEEE Transactions on Industrial Informatics,* vol. 10, pp. 351-363, 2014.

[71] L. D. Mendes and J. J. Rodrigues, "A survey on cross-layer solutions for wireless sensor networks," *Journal of Network and Computer Applications,* vol. 34, pp. 523-534, 2011.
[72] G. Miao, N. Himayat, Y. G. Li, and A. Swami, "Cross-layer optimization for energy-efficient wireless communications: a survey," *Wireless Communications and Mobile Computing,* vol. 9, pp. 529-542, 2009.
[73] M. Di Francesco, S. K. Das, and G. Anastasi, "Data collection in wireless sensor networks with mobile elements: A survey," *ACM Transactions on Sensor Networks (TOSN),* vol. 8, p. 7, 2011.
[74] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava, "Optimizing sensor networks in the energy-latency-density design space," *IEEE Transactions on mobile computing,* vol. 99, pp. 70-80, 2002.
[75] L. Bölöni and D. Turgut, "Should I send now or send later? A decision-theoretic approach to transmission scheduling in sensor networks with mobile sinks," *Wireless Communications and Mobile Computing,* vol. 8, pp. 385-403, 2008.
[76] A. Kansal, A. A. Somasundara, D. D. Jea, M. B. Srivastava, and D. Estrin, "Intelligent fluid infrastructure for embedded networks," in *Proceedings of the 2nd international conference on Mobile systems, applications, and services*, 2004, pp. 111-124.
[77] G. Anastasi, M. Conti, E. Gregori, C. Spagoni, and G. Valente, "Motes sensor networks in dynamic scenarios: an experimental study for pervasive applications in urban environments," *Journal of Ubiquitous Computing and Intelligence,* vol. 1, pp. 9-16, 2007.
[78] L. Pelusi, A. Passarella, and M. Conti, "Encoding for Efficient Data Distribution in Multihop Ad Hoc Networks1," *Algorithms and Protocols for Wireless and Mobile Ad hoc Networks,* p. 87, 2009.
[79] K. Dantu and G. S. Sukhatme, "Connectivity vs. control: Using directional and positional cues to stabilize routing in robot networks," in *Robot Communication and Coordination, 2009. ROBOCOMM'09. Second International Conference on*, 2009, pp. 1-6.

[80] K. Akkaya and M. Younis, "Energy-aware routing to a mobile gateway in wireless sensor networks," in *Global Telecommunications Conference Workshops, 2004. GlobeCom Workshops 2004. IEEE*, 2004, pp. 16-21.
[81] A. A. Somasundara, A. Kansal, D. D. Jea, D. Estrin, and M. B. Srivastava, "Controllably mobile infrastructure for low energy embedded networks," *IEEE Transactions on Mobile Computing,* vol. 5, pp. 958-973, 2006.
[82] S. A. Awwad, C. K. Ng, N. K. Noordin, and M. F. A. Rasid, "Cluster based routing protocol for mobile nodes in wireless sensor network," in *Collaborative Technologies and Systems, 2009. CTS'09. International Symposium on*, 2009, pp. 233-241.
[83] C. Perkins and I. Chakeres, "Dynamic MANET on-demand (DYMO) routing," *draft-ietf-manet-dymo-26 (work in progress),* p. 127, 2013.
[84] S. Gao, H. Zhang, and S. K. Das, "Efficient data collection in wireless sensor networks with path-constrained mobile sinks," *IEEE Transactions on Mobile Computing,* vol. 10, pp. 592-608, 2011.
[85] S. Mohseni, R. Hassan, A. Patel, and R. Razali, "Comparative review study of reactive and proactive routing protocols in MANETs," in *Digital ecosystems and technologies (DEST), 2010 4th IEEE international conference on*, 2010, pp. 304-309.
[86] D. Johnson, Y.-c. Hu, and D. Maltz, "The dynamic source routing protocol (DSR) for mobile ad hoc networks for IPv4," 2070-1721, 2007.
[87] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc on-demand distance vector (AODV) routing," 2070-1721, 2003.
[88] V. D. Park and M. S. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," in *INFOCOM'97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution., Proceedings IEEE*, 1997, pp. 1405-1413.
[89] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile

computers," in *ACM SIGCOMM computer communication review*, 1994, pp. 234-244.

[90] S. A. Mohammad, A. Rasheed, and A. Qayyum, "VANET architectures and protocol stacks: a survey," in *International Workshop on Communication Technologies for Vehicles*, 2011, pp. 95-105.

[91] T. L. Willke, P. Tientrakool, and N. F. Maxemchuk, "A survey of inter-vehicle communication protocols and their applications," *IEEE Communications Surveys & Tutorials,* vol. 11, 2009.

[92] I. Bekmezci, O. K. Sahingoz, and Ş. Temel, "Flying ad-hoc networks (FANETs): A survey," *Ad Hoc Networks,* vol. 11, pp. 1254-1270, 2013.

[93] J. H. Forsmann, R. E. Hiromoto, and J. Svoboda, "A time-slotted on-demand routing protocol for mobile ad hoc unmanned vehicle systems," in *Defense and Security Symposium*, 2007, pp. 65611P-65611P-11.

[94] B. Karp and H.-T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th annual international conference on Mobile computing and networking*, 2000, pp. 243-254.

[95] A. I. Alshabtat, L. Dong, J. Li, and F. Yang, "Low latency routing algorithm for unmanned aerial vehicles ad-hoc networks," *International Journal of Electrical and Computer Engineering,* vol. 6, pp. 48-54, 2010.

[96] S. Chaumette, R. Laplace, C. Mazel, R. Mirault, A. Dunand, Y. Lecoutre*, et al.*, "Carus, an operational retasking application for a swarm of autonomous uavs: First return on experience," in *MILITARY COMMUNICATIONS CONFERENCE, 2011-MILCOM 2011*, 2011, pp. 2003-2010.

[97] W. Huba and N. Shenoy, "Airborne surveillance networks with directional antennas," in *IARIA International conference on Computers and network Systems, ICNS*, 2012.

[98] A. I. Alshbatat and L. Dong, "Cross layer design for mobile ad-hoc unmanned aerial vehicle communication networks," in

Networking, Sensing and Control (ICNSC), 2010 International Conference on, 2010, pp. 331-336.
[99] O. N. Challa, "CubeSat Cloud: A framework for distributed storage, processing and communication of remote sensing data on cubesat clusters," 2013.
[100] O. N. Challa and J. McNair, "Cubesat torrent: Torrent like distributed communications for cubesat satellite clusters," in *MILCOM 2012-2012 IEEE Military Communications Conference*, 2012, pp. 1-6.
[101] O. Challa and J. McNair, "Distributed Computing on CubeSat Clusters using MapReduce," in *Proceedings of the 1st Interplanetary CubeSat Workshop, Cambridge, MA*, 2012.
[102] O. N. Challa and J. McNair, "Distributed Data Storage on CubeSat Clusters," *Advances in Computing,* vol. 3, pp. 36-49, 2013.
[103] P. P. Sundaramoorthy, E. Gill, and C. Verhoeven, *Systematic Identification of Applications for a Cluster of Femto-satellites*: International Astronautical Federation, 2010.
[104] M. de Milliano and C. Verhoeven, "Towards the next generation of nanosatellite communication systems," *Acta Astronautica,* vol. 66, pp. 1425-1433, 2010.
[105] P. Rodrigues, A. Oliveira, R. Mendes, S. Cunha, R. Garcia Von Pinho, C. Salotto*, et al.*, "GAMANET: Disrupting communications and networking in space," presented at the 64th International Astronautical Congress, Beijing, China, 2013.
[106] K. Sidibeh and T. Vladimirova, "Wireless communication in LEO satellite formations," in *Adaptive Hardware and Systems, 2008. AHS'08. NASA/ESA Conference on*, 2008, pp. 255-262.
[107] R. Radhakishnan, W. Edmonson, and Q. Zeng, "The performance evaluation of distributed inter-satellite communication protocols for cube satellite systems," in *The 4th Design, Development and Research Conference, Capetown, South Africa*, 2014.
[108] R. Sun, J. Guo, E. Gill, and D. Maessen, "Potentials and limitations of CDMA networks for combined inter-satellite

communication and relative navigation," *Int J Adv Telecommun,* vol. 5, 2012.

[109] B. Chen and L. Yu, "Design and implementation of LDMA for low earth orbit satellite formation network," in *Embedded and Ubiquitous Computing (EUC), 2011 IFIP 9th International Conference on*, 2011, pp. 409-413.

[110] W. Alliance, "WiMedia logical link control protocol," *WLP Specification Approved Draft,* vol. 1, 2007.

[111] R. Radhakrishnan, W. W. Edmonson, F. Afghah, J. Chenou, R. M. Rodriguez-Osorio, and Q.-A. Zeng, "Optimal multiple access protocol for inter-satellite communication in small satellite systems," in *4S Small Satellite Systems and Services Symposium*, 2014.

[112] E. Ekici, I. F. Akyildiz, and M. D. Bender, "Network layer integration of terrestrial and satellite IP networks over BGP-S," in *Global Telecommunications Conference, 2001. GLOBECOM'01. IEEE*, 2001, pp. 2698-2702.

[113] I. F. Akyildiz, E. Ekici, and G. Yue, "A distributed multicast routing scheme for multi-layered satellite IP networks," *Wireless Networks,* vol. 9, pp. 535-544, 2003.

[114] M. A. Bergamo, "High-Throughput Distributed Spacecraft Network: architecture and multiple access technologies," *Computer Networks,* vol. 47, pp. 725-749, 2005.

[115] C.-T. Cheng, K. T. Chi, and F. C. Lau, "An energy-aware scheduling scheme for wireless sensor networks," *IEEE Transactions on vehicular technology,* vol. 59, pp. 3427-3444, 2010.

[116] D. Gerhardt, M. Bisgaard, L. Alminde, R. Walker, M. A. Fernandez, A. Latiri, *et al.*, "GOMX-3: Mission Results from the Inaugural ESA In-Orbit Demonstration CubeSat," 2016.

[117] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, *et al.*, "Delay-tolerant networking: an approach to interplanetary internet," *IEEE Communications Magazine,* vol. 41, pp. 128-136, 2003.

[118] D. J. Mudgway and R. Launius, *Uplink-Downlink: A History of the Deep Space Network, 1957-1997*, 2001.
[119] L. F. Van Hoesel and P. J. Havinga, "A lightweight medium access protocol (LMAC) for wireless sensor networks: Reducing preamble transmissions and transceiver state switches," 2004.
[120] F. N. Ali, P. K. Appani, J. L. Hammond, V. V. Mehta, D. Noneaker, and H. Russell, "Distributed and adaptive TDMA algorithms for multiple-hop mobile networks," in *MILCOM 2002. Proceedings*, 2002, pp. 546-551.
[121] W. T. Webb and L. Hanzo, *Modern Quadrature Amplitude Modulation: Principles and applications for fixed and wireless channels: one*: IEEE Press-John Wiley, 1994.
[122] S. M. Bilal, C. J. Bernardos, and C. Guerrero, "Position-based routing in vehicular networks: A survey," *Journal of Network and Computer Applications,* vol. 36, pp. 685-697, 2013.
[123] S. Ennis, "CubeSat Networks: Balancing Power with Satellite-to-Ground Data Throughput " *YouTube Video,* 13/5/2017 2017.
[124] S. Ennis, "CubeSat Networks: Balancing Power with Satellite-to-Ground Data Throughput," *GitHub repository,* 2017.
[125] T. Steinbach, H. D. Kenfack, F. Korf, and T. C. Schmidt, "An extension of the OMNeT++ INET framework for simulating real-time ethernet with high accuracy," in *Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques*, 2011, pp. 375-382.
[126] G. F. Riley and T. R. Henderson, "The ns-3 network simulator," *Modeling and tools for network simulation,* pp. 15-34, 2010.
[127] H. Klinkrad, *Space debris*: Wiley Online Library, 2010.
[128] D. Oltrogge and K. Leveque, "An evaluation of CubeSat orbital decay," 2011.
[129] J. Andrews, K. Watry, and K. Brown, "Nanosat deorbit and recovery system to enable new missions," 2011.
[130] B. Cohen, "Lunar Flashlight: mapping lunar surface volatiles using a CubeSat," 2014.

[131] F. Ferrari, M. Lavagna, B. Burmann, I. Gerth, M. Scheper, and I. Carnelli, "ESA's Asteroid Impact Mission: Mission Analysis and Payload Operations State of the Art," 2016.