CubeSat Networks: The Power Throughput Trade-Off

Stephen Ennis

Supervisor: Jonathan Dukes

School of Computer Science and Statistics

Trinity College Dublin

M.A.I. Computer Engineering

18/05/2017

Declaration

...

Permission Lend and/or Copy

...

Acknowledgements

I would like to thank Dr. Jonathon Dukes for his exceptional supervision and guidance throughout my dissertation. Particular thanks are due as Dr. Dukes took on the supervision of my dissertation project voluntarily during a notably busy period of the academic term. Working with Dr. Dukes has proved, from the offset, 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 countless engineers, scientists, operators and researchers behind missions such as NASA's Nodes and EDSN missions, ESA's GomX-3 mission and the Shanghai Engineering Center for Microsatellites's 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, whose continual hard work and sacrifice has supported me throughout my studies.

Abstract

...

Contents

Figures

Tables

Abbreviations

LEO	Low Earth Orbit
SMB	Small to Medium Business
…	…

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 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's unit cost to LEO, with a fully loaded cargo bay, was 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 reusability (Figure 1).
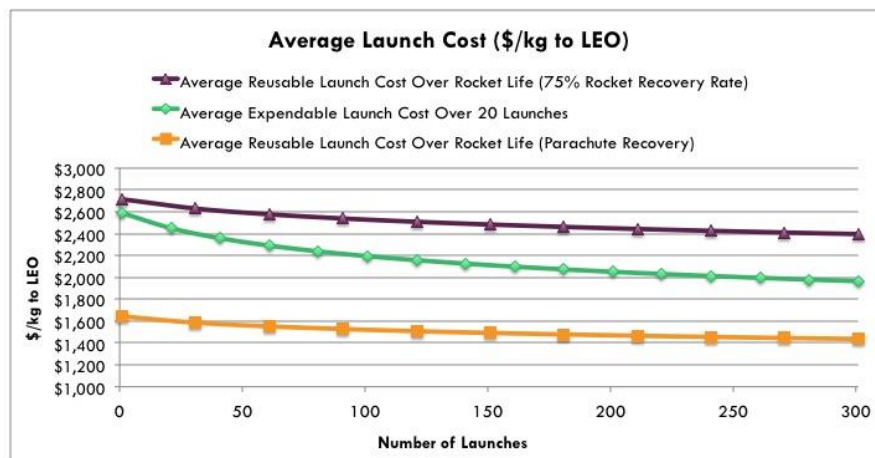
Figure 1 A projection of unit costs to LEO as the number of launches of a particular vehicle type increases. Image Credit: ARK Investment Management LLC.

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 factor of CubeSats is 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 [3]. Such open-sourcing is historically rare in the satellite industry.
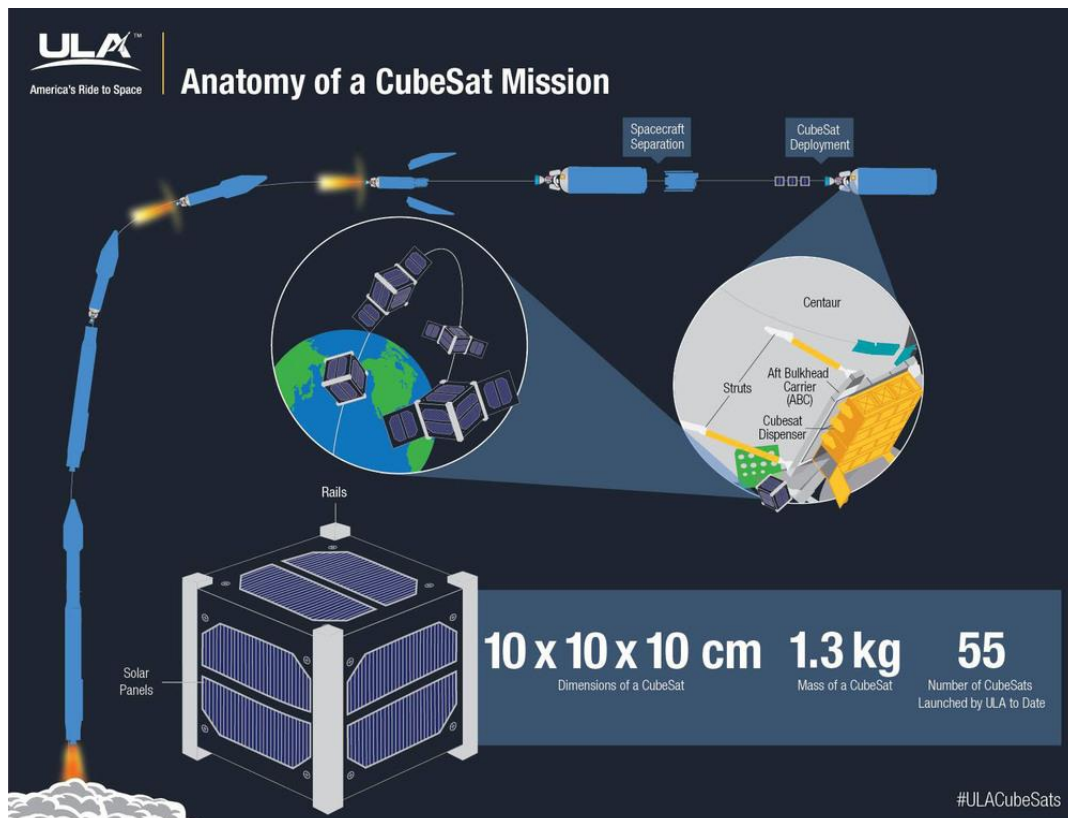
Figure 2. An illustration of the CubeSat form factor and a deployment approach. Here CubeSats are deployed prior to delivery of the primary payload using a dispenser attached to the final stage of the launch vehicle. Image Credit: United Launch Alliance LLC.

CubeSats, as the name suggests, adopt a cube form factor. Each Cube, often referred to as a 'unit', is 10cm to a side (Figure 2). Multiple units are often combined in order to form larger CubeSats. Six unit configurations are typically the largest form factor used [4, 5]. 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 [6], attitude determination and control (ADCS) [7], communications [8-10], and command and data handling (C&DH) [11, 12]. Alongside its sub-systems, a CubeSat often carris a small 'payload' which may be a scientific instrument or some previously 'unflown' component such as an experimental antenna [13]. 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 the educational and academic domains [14].

CubeSats, and other small satellites, have an advantage over larger satellites in their ability to 'hitch' a ride alongside primary launch payloads. Effectively all modern large 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 have been developed which can make use of this spare volume and lift capacity [15, 16]. Such deployers can often carry multiple CubeSats. 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 increasing ambitious as a result of the lowering unit costs to LEO and the increasing affordability and capabilities of COTS CubeSat components [17-19]. This project focuses on a particular 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 [20], high fidelity sensory data, inexpensive low-data rate terrestrial communications and improved air traffic monitoring [21]. The space industry has taken the first crucial steps into designing and testing CSNs with missions such as EDSN (Edison Demonstration of Smallsat Networks) [22] (Figure 3), Nodes (Network & Operation Demonstration Satellites) [12] and Tianwang-1 [23].
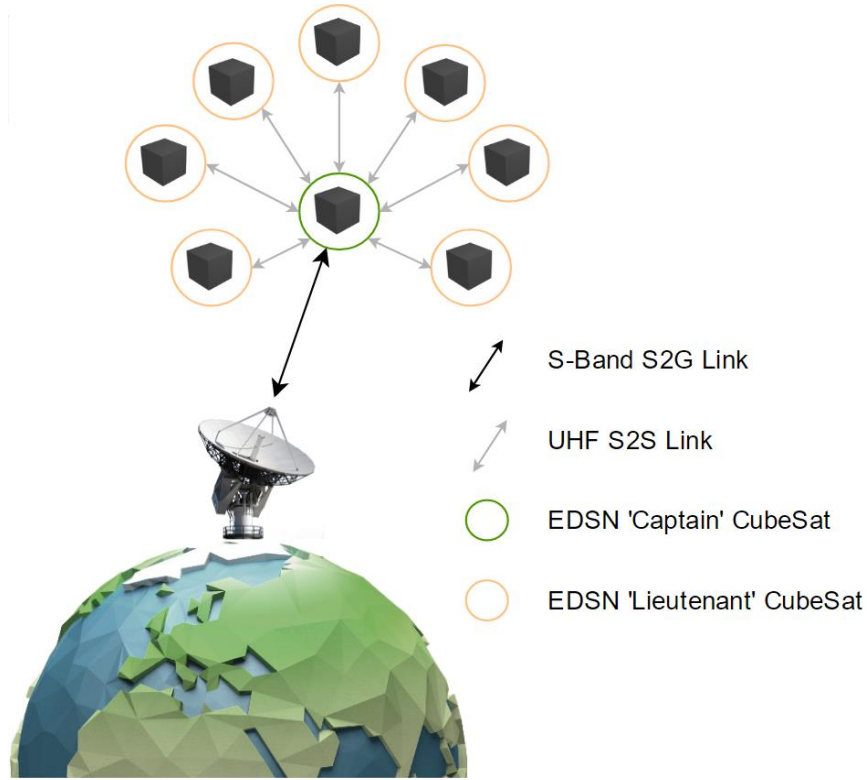
Figure 3. An illustration of EDSN CubeSats in orbit forming a star (hub-and-spoke) topology CSN. Each CubeSat houses two radios, one for S2S the other and S2G communication. Communication to ground is perform only be the current elected 'Captain' CubeSat.

This work seeks to build upon data from the aforementioned missions. The overall aim being the exploration of fundamental aspects 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

CSNs share many similarities with terrestrial concepts such as wireless sensor networks (WSNs) and mobile ad-hoc networks (MANETs). The state of the art in CSNs has its basis

in works from these fields. Much of the existing work relating directly to CSNs was published prior the design and launch of the first CSN mission. As a result, authors were often forced to make several assumptions as to the capabilities and constraints of CSNs.

The general motivation of this work is to assess CSN network and MAC layer protocol design. This assessment requires the analysis of existing work relating to WSNs, MANETs, CSNs, and the design and implementation of CubeSat missions. As mentioned, there are numerous and varied applications of CSNs. This work narrows the scope of interest to a hypothetical mission, which is detailed further in section 1.2.

In the hypothetical mission, each CubeSat is assumed to carry a scientific instrument which produces some data which must be communicated to ground. The overall performance of the CSN's S2G communication may be assessed from the quality, quantity, and/or timeliness of the data received. It is assumed that issues relating to data quality are fully addressed. This leaves the quantity and timeliness of data received as metrics for the success of this hypothetical mission. The timeliness of data reception is important in applications such as communications and real-time Earth Observation (EO). This work makes an assumption that the value of science data is not affected by the time taken for said data to reach ground, or, for that matter, by the order in which data arrives. 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, satellite-to-satellite S2S is generally considerably 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. These imbalances differentiate CSN networks from many similar terrestrial networks. Increasing the amount of S2G communication will increase S2G throughput however, it will also consume more energy

6

overall and reduce the mission's lifetime. A reduction in mission lifetime will directly decrease its maximum possible throughput through reduced data collection. 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. Any solutions proposed by this work intend to address the trade-offs which arise when attempting to balance energy consumption with S2G throughput.

This work presents simulations of modified communications protocols carried out using the discrete event network simulator OMNeT++ [24]. These protocols were chosen and implemented with consideration to the aforementioned power versus throughput (PvTP) trade-off for CSNs. The optimal approach to the PvTP trade-off is that which enables the greatest throughput while minimizing energy consumption (power use). 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. Ultimately, this work's proposed protocols are assessed on their ability to increase throughput without compromising on overall energy consumption.

The PvTP trade-off is affected by numerous mission aspects such as a CubeSat's energy storage, battery recharging, and communications capabilities. These capabilities will vary from mission to mission and are liable to develop significantly over the coming years. As far as is possible, CubeSat capabilities are modelled on the current state of the art. Advancements in CubeSat technology may alter the context of the PvTP trade-off. However, the core trade-off of throughput at the cost of energy consumption will remain.

1.2 Hypothetical Mission
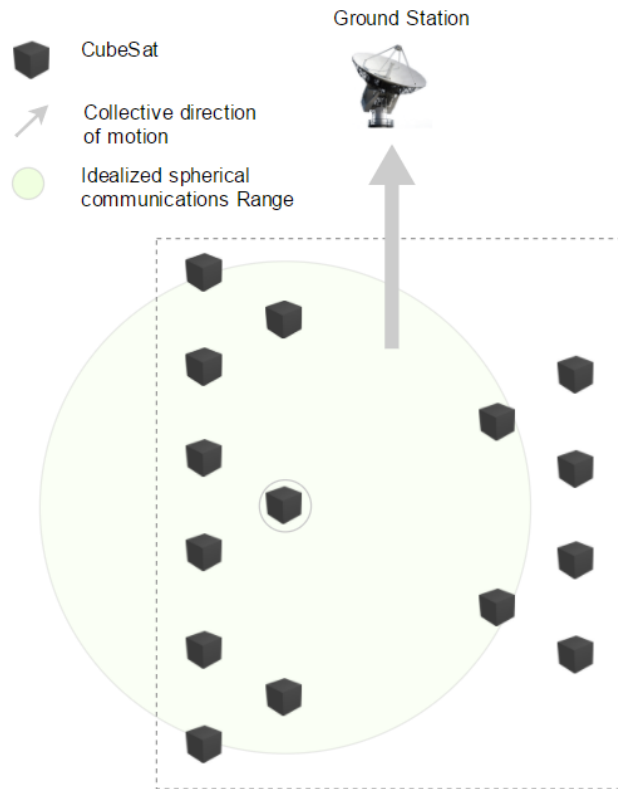
Figure 4. The CubeSats and ground station of the hypothetical as viewed from a higher orbit looking down upon the Earth's surface. The CubeSats are assumed to have an orbital altitude of 500km.

Section 1.1 introduces the general aspects of this work's chosen hypothetical mission. Alongside the chosen PvTP trade-off problem, the assumptions made regarding the hypothetical mission significantly influence this work's focus and decision making. The design of 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 4)

- The speed, direction and altitude of each CSN's ground pass remain constant (Figure 4)

- 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 to the ground station

- CubeSat radios have ideal spherical communication ranges which are unaffected by the craft's orientation (Figure 4)

- Only one ground station is available (Figure 4)

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 design.

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

the Chapter 4. This chapter presents in-depth explanations and illustrations of the proposed protocols alongside justifications of relevant design choices.

Chapter 4, "Simulation", deals primarily with the experimental setup, implementation and analysis of the protocols simulated using OMNeT++. Detailing the experimental setup involves discussing several assumptions made regarding CubeSat and space-bound wireless communications. This chapter also places a focus on the challenges relating to 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 detailing potential future work in the development of the proposed protocols and in simulating CubeSat communications.

Chapter 2: State of the Art

The major literature informing this work can be roughly 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 examination of relevant terrestrial communication technologies which focuses on Wireless Sensor Networks (WSNs) and Mobile Ad-Hoc Networks (MANETs). Detailing relevant terrestrial technology provides important context for the following section on CubeSat communications. This final major section seeks to explore, in depth, examples of the latest proposed approaches to CubeSat communications. Finally several notable secondary areas of research are 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 [25]. The model is used to conceptually separate various aspects of network communications into a number of 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 performing packet addressing, sequencing and routing operations. Entitles in the Data Link layer perform error correction and manage access to shared communication media. Elements 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 within the Data Link layer and the other within the Network layer.

This chapter covers 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 justify or challenge this work's design decisions.

2.1 CubeSats

CubeSats typically fall within the 'Nano' satellite weight class (1-10kg). CubeSats are further classified by the number of 'units' which they contain, where a unit is a 10cm cube. A one unit CubeSat is referred to as a '1U'. Configurations of 1, 2, 2.5, 3 and 6U are all common.

CubeSats were first proposed by Bob Twiggs of Stanford University and Jordi Puig-Suari of California Polytechnic State University in 1999 [26]. In 2000 the first work detailing a new CubeSat "standard" was published [27]. CubeSats were proposed to address the prohibitive costs and challenges involved in satellite development for academic purposes. At the time, there were effectively no standard approaches or components for the design and implementation of small satellites. Researchers relied almost entirely on placing instruments, alongside primary payloads, on larger satellites or pursuing the development of research satellites as lengthy collaborations across multiple research institutions. This was despite the fact that, frequently, research only required satellites with basic capabilities.

In 2003 the CubeSat was launched on-board a Russian Eurorockot [26]. At the time of writing, May 2017, there have been 487 CubeSats successfully launched or deployed into orbit [28]. Spread across 14 years this number may seem unimpressive, however, approximately 75% of all these launches took place after 2011. 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 [29], roughly 40% of all CubeSats were 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, and the remaining 10% is divided between various commercial, military and scientific applications [28].

Unsurprisingly, the core motivation behind the recent popularity of CubeSat missions is cost. Costs are driven down by three factors, the use of COTS components, open sourcing, and low launch expsenses. Effectively every component of a modern CubeSat is available in COTS form. Retailers such as Clyde Space Ltd. offer a wide range of products from batteries to attitude determination and control systems [30]. 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 systems are becoming increasingly widespread as the platform develops [3]. 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 Puig-Suari at the California Polytechnic State University (CalPoly), have advanced the domain through the specification of a number of pseudo-standards. Crucially, researchers at CalPoly led the development and design of standard CubeSat deployers [31] (Figure 5). Such deployers have come to define de-facto standards for the domain. 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 [32]. Recently, "OpenOrbiter" by Straub et al. from the University of North Dakota is a prime example of an open pseudo-standard framework for CubeSat development [33].

Figure 5. Three 1U CubeSats beside a 3U (oly 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 the CubeSat, development costs may range anywhere from $50,000 to $250,000 [34]. This can be compared to a development cost in the order of millions of dollars for larger 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

14

of thousands, if not millions, of dollars. This of course depends heavily on the launch vehicle used and the orbital requirements (Low, High, Medium, Solar etc.). CubeSats avoid these costs by 'hitching' a ride as secondary payloads through the use of volume and lift capacity not required by primary payloads. Providers such as SpaceX have disrupted the satellite industry by offering access to LEO at considerably reduced costs [35]. These factors have led to CubeSat launch costs as low as $10,000 [34]. With recent development towards multi-CubeSat and CSN missions, multiple CubeSats may take the place of their larger counterparts at a fraction of the cost. CubeSats are also making the move beyond LEO with new developments towards deep space and lunar applications [16, 18] (Figure 6).
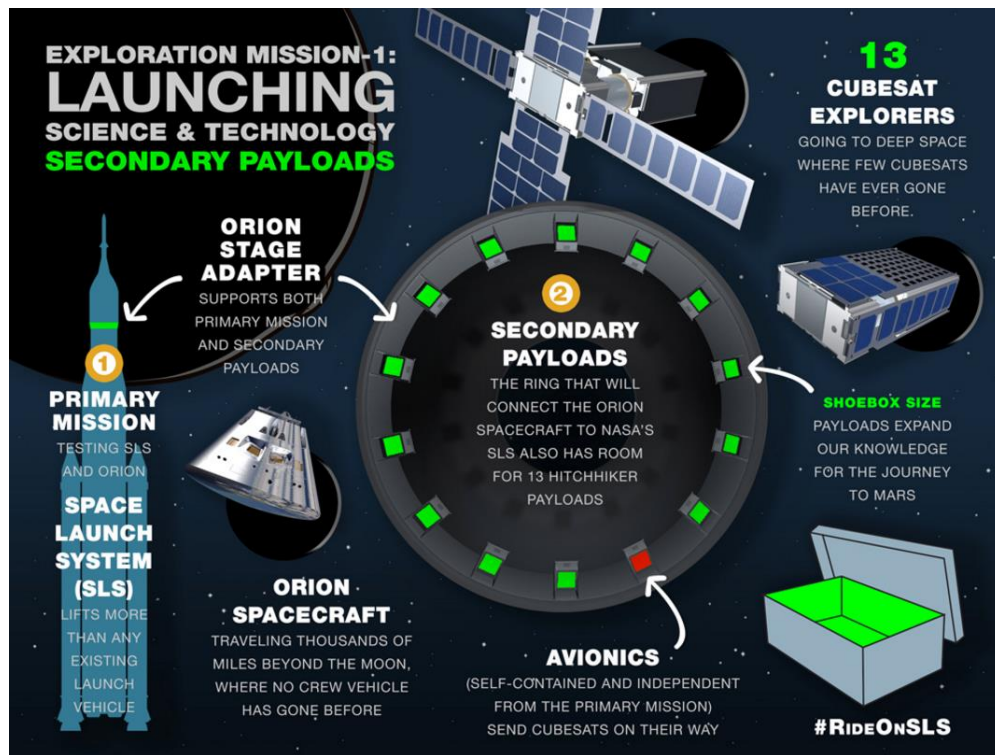
Figure 6. An illustration of the dedicated secondary payload deployers built into the "Orion Stage Adapter" of NASA's upcoming Space Launch System. Existing launch vehicles are generally retroactively fitted with such deployers. SLS is projected to launch the first CubeSat into deep space in 2019. Image Credit: NASA.

15

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. In many cases, there is a need to carefully assess potential communications strategies in the context of the current and emerging capabilities of the CubeSat platform.

Satellite-to-Ground Communication Systems

There is considerable variance in the implementation of S2G CubeSat communication sub-systems. The choice and design of such system is application dependent however, there are some broad patterns worth noting. For instance, the most common protocol for S2G communications is AX.25 [36]. Implementations using AX.25 at the link layer generally utilize UDP and IP protocols at higher protocol layers. CubeSat S2G communications sub-systems typically 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 achieved a remarkable S2G maximum data rate of 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 were chosen. One of the primary inspirations for these characteristics is the Tianwang-1 (TW-1) mission [40]. This mission is an ideal candidate to use as a baseline for S2G communication modelling as the mission was designed specifically to test CubeSat inter-communications. The mission is representative of the capabilities future CSN missions may achieve. TW-1 achieved S2G data rates of 125kbps. Details regarding the power consumption of the TW-1 S2G sub-systems are unavailable. However, by examining previous and upcoming missions as well as work on energy budget analysis one may assume a peak transmission power consumption of 3W [41].

Satellite-to-Satellite Communication Systems

CubeSat S2S communication is by no means a new concept. The field began to gain popularity following a paper 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 2016 when the NASA Nodes mission made it a reality [12].

Unlike the case of S2G communications, there is comparatively little prior art regarding the S2S capabilities of CubeSats. This is unsurprising considering the age of the domain. There are effectively two cases which inform the state of the art of S2S CubeSat capabilities: The NASA's Nodes mission and Gamalink [42]. As will be discussed, Gamalink is considerably more advanced than the systems employed on the Nodes mission. Nodes utilized a UHF transceiver and the AX.25 protocol to achieve S2S data

rates of 12kbps. As Nodes was a first in many regards it is unsurprising that the mission designers opted for a basic approach to CubeSat communications.

Gamalink is a proprietary SDR based technology developed by Tekever, a Portuguese Aerospace and Defence company. Gamalink is unquestionably the current state of the art in CubeSat S2S communications. Gamalink has been successfully tested on the TW-1 mission [23] it is also marked for use by several other missions such as i-INSPIRE II [43], DelFFi [44] and ESA's Proba 3 [45].
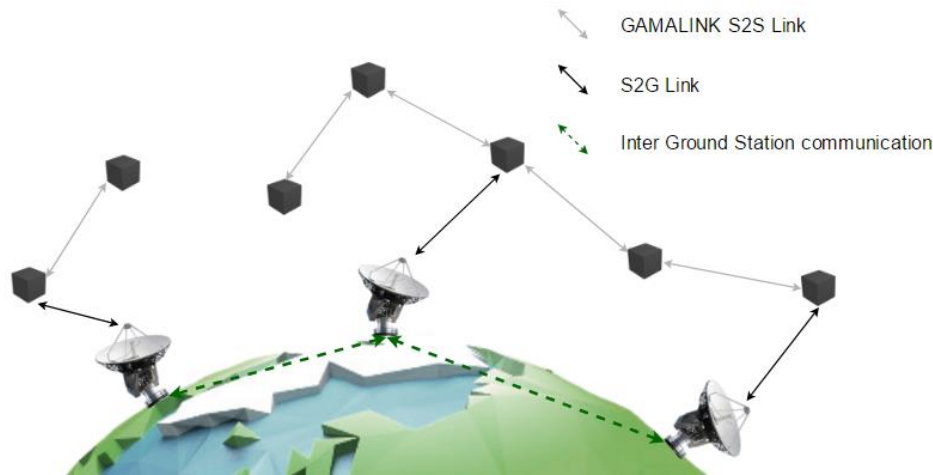
Figure 7. Unlike NASA's EDSN approach (Figure 3), Gamalink seeks to establish multi-hop CubeSat networks capable of communicating with multiple ground stations. Gamalink designers refer to such networks as GAMANETs.

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

18

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 the maximum data rate as 1Mbps [43]. Gamalink 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]. Using these details and assuming a maximum data rate of 2Mbps, a baseline can be established for this work's simulation of CubeSat S2S communications.

Battery and Recharge Capabilities

The energy storage and recharging capabilities of CubeSats varies considerably from mission to mission. The form factor employed for a given CubeSat determines the maximum volume available to house batteries and the surface area available for solar arrays. Folding solar panels are common place on larger spacecraft. It follows that folding solar panels have also been investigated for use on CubeSat missions [46]. NASA's EDSN mission is used as the an 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 its 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 (~47.5m). 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 worthnoting. 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 craft will also be required to 'de-tumble' prior to launch. There are numerous tested examples of ADCS technologies for CubeSats [7, 47, 48] several of which are available COTS (Figure 8). Along with ADCS some basic orbital control and manoeuvrability systems have also been tested at the CubeSat scale [6]. Such systems provide CubeSats with the basic capabilities to maintain regular orbits and formations.

Figure 8. A COTS CubeSat attitude control unit. The rotational velocity of the three reaction wheels shown above can be altered in order to adjust attitude. Image Credit: Clyde Space Ltd, All Rights Reserved.

The determination of accurate time and position are two classic challenges for spacecraft that have also been solved in the domain of CubeSats. By communicating with Global Navigation Satellite System (GNSS) a CubeSat may acquire precise GPS time, velocity and position. 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 the aforementioned Gamalink technology provides functionality beyond that of S2S communications. 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. For the purposes of this work, references to CubeSat 'applications' should be considered synonymous with CubeSat mission objectives. 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. 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 is a simplified multi-CubeSat mission. Each CubeSat is assumed to carry some sensing instrument. This instrument is considered to be a black box which performs some sensing and produces some data.

Two recent sensing missions are worth detailing: 3Cat-2 [50] and RAVAN (Radiometer Assessment using Vertically Aligned Nanotubes) [51]. 3Cat-2 is 6U CubeSat developed at the Universidad Politécnica de Cataluña. It was launched in August of 2016 on-board a CZ-2D (Chang Zheng-2D) operated by the CNSA. 3Cat-2 represents a significant mission in the state of the art for Earth Observation (EO) missions. The extensive use of GNSS based systems also make 3Cat-2 an interesting case for the application of CSNs. 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 the baseline chosen for S2G communications.

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 signals off of 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 multi-CubeSat mission. However, 3Cat-2's active sensing is uniquely suited to adaption for 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 for CSNs.

RAVAN is a 3U CubeSat developed at the Johns Hopkins Applied Physics Laboratory. RAVAN was launched November of 2016 aboard an Atlas-5 as part of NASA's ELaNa (Educational Launch of Nanosatellites) program. RAVAN is highly relevant to this work as the 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 9). Unfortunately, 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.
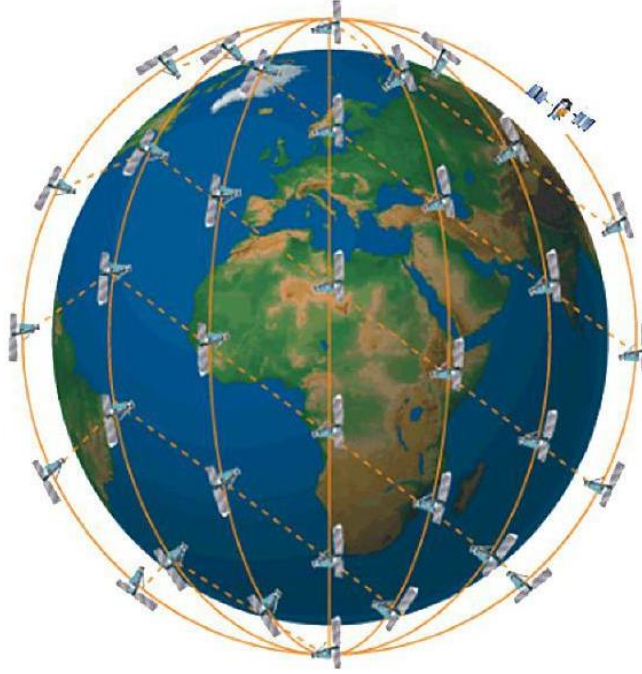
Figure 9. A conceptual illustration of the proposed RAVAN constellation.
Image Credit: John Hopkins University Applied Physics Laboratory.

It should be noted that CubeSat sensing is a deep and broad field. The examples of RAVAN and 3Cat-2 are in no way intended to illustrate a comprehensive study of the domain. Notable upcoming missions such as CeREs (a Compact Radiation belt Explorer) [53], LAICE (Lower Atmosphere/Ionosphere Coupling Experiment) [54], and SOCON (Sustained Ocean Observation from Nanosatellites) [55] seek to advance the already diverse sensing applications of CubeSats. In several regards, it is these advanced applications which have driven interest in CSN based missions. As applications develop, so too does interest in advancements such as multi-point measurements, in-orbit interferometry [56] and synchronized observation. These advancements call for the introduction of CSNs. CSNs are, in many regards, an obvious next step in the development of CubeSat sensing missions.

CubeSat Network Missions

There are three major missions to consider 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 we 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 autonomous communication and coordination of CubeSat's in orbit. Each ESDN craft is a 1.5U CubeSat weighing ~1.73kg. A Samsung® smartphone provides activity scheduling and execution for each craft. Several secondary COTS micro-controllers handle various activities involving GNSS communication, command and data handling, scientific measurements, ADCS and sensor input handling. The 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 an in-orbit CubeSat network, it is clear that 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. These works state which COTS components were used in the construction of EDSN craft. 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 North American Aerospace Defence Command (NORAD)). The AstroDev transceiver enables S2S communications at 9.6kbps and uses the AX.25 protocol at the data link layer. The Nodes mission scaled this data rate back to 1.2kbps, potentially to increase the 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. The link layer, as detailed by the OSI reference model [25], is concerned primarily with medium access control (MAC). 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 3). 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 10). 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 pings 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 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 then proceed to ping each remaining Lt in a fixed order (Figure 10). The Cpt will wait up to four minutes for a response from a pinged Lt before giving up and moving on to ping the next Lt.
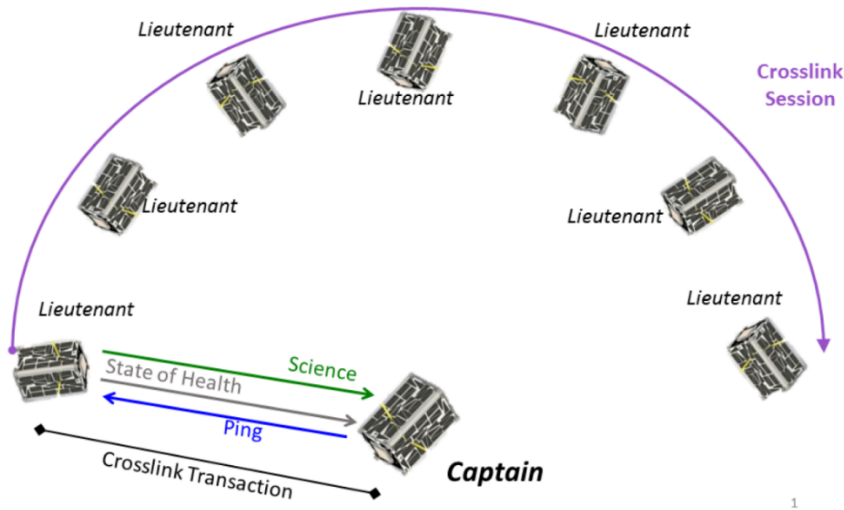
Figure 10. The Cpt/Lt protocol. EDSN designers refer to S2S communication as crosslinking. The Captain pings a Lieutenant before receiving state-of-health and science data packets. Image Credit: NASA Ames Research Centre

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 GPS time in order to determine whether to assume the role of Cpt. If a craft cannot get GPS time it does not participate in either S2S or S2G communications. The duration a craft holds the Cpt role is referred to as a "minor cycle". Each minor cycle lasts roughly 25 hours and includes 3-4 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 GPS 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. Seven minor cycles, one for each Lt, come together to form a major cycle. A major cycle lasts for roughly eight days. 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 sessions follow a fixed schedule. A Lt will only begin listening for Cpt pings at a predetermined time during each minor cycle. During a minor cycle each craft updates it's GPS time, position and velocity only once. Mission designers predict the maximum relative clock drift between a Lt and the Cpt to be 12 seconds. As such, a Lt will begin listening for Cpt pings 30 seconds before the scheduled time and will continue listening 30 seconds after the expected sixth and final ping. Each session start time is 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. Effectively all aforementioned salient aspects of the EDSN mission remain. Despite only involving two craft, Nodes was able to achieve many of the objectives of EDSN. A number of small changes were made to mission objectives and there were some notable changes to the

custom Cpt/Lt communication protocol. These changes are detailed in a work published in 2016 following the mission's successful launch, deployment and conclusion [12].

Where EDSN focussed purely on S2G communication, Nodes advances one step further by introducing to demonstration of Ground-to-Space (G2S) remote commands. In Nodes, the 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 with science and state-of-health packets, command packets are implemented along with specialized command acknowledgements and responses. Although this work focuses purely on S2G communications, it is worth noting that Nodes was the first demonstration of indirect command and control of CubeSats using CubeSat S2S communication.

There are several notable changes to the Cpt/Lt protocol introduced in the Nodes mission. Firstly, 12 pings over 110 seconds are used rather than 6 pings over 50 seconds. Rather than a fixed order of minor-cycles, the captaincy is negotiated between the Nodes craft (Figure 11). A default Cpt craft compares metrics relating to battery voltage, 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 manages the negotiation. 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 demote itself to the Lt 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.
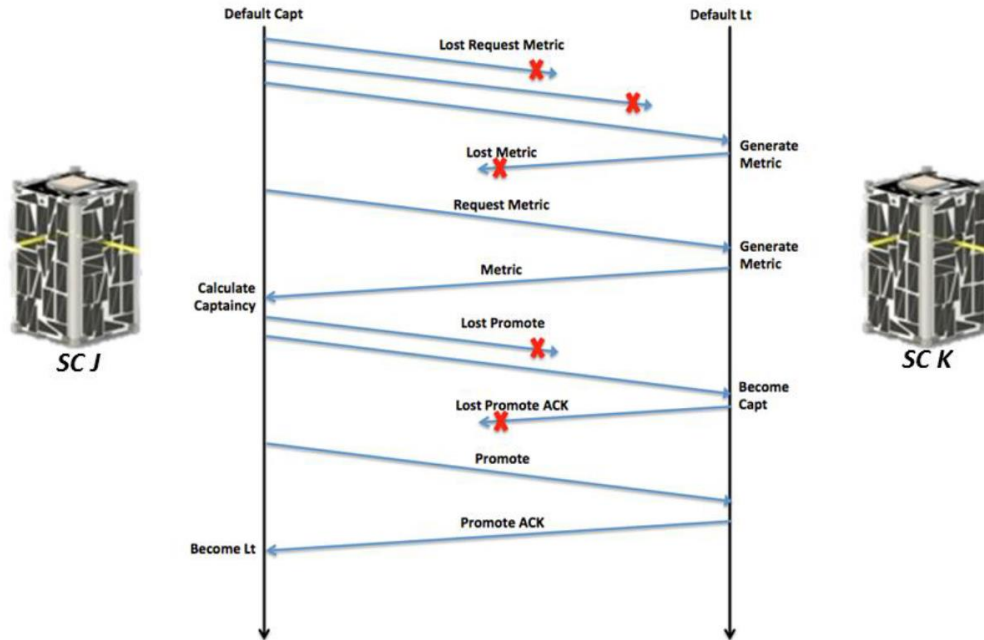
Figure 11. A timeline of the Captaincy negotiation process carried out between the two Nodes spacecraft (SC). ACK: Acknowledgement. Image Credit: NASA Ames Research Centre

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 session

- One successfully executed indirect command

- Two successful captaincy negotiations

- The collection and receipt of 20 days of spacecraft state-of-health.

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

30

Nodes mission designers lay out numerous desirable enhancements such as: improved clock synchronization, inter-sat ranging, multi-hop routing, further acknowledge systems and/or 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 (TW1) mission, also referred to as STU-2, was a three CubeSat CSN mission involving numerous commercial and academic entities led by the Shanghai Engineering Centre for Microsatellites (SECM). The majority of the published work relating to TW1 details its ADCS and novel 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, the details of the proprietary Gamalink technology are closely guarded. For known details on Gamalink see section 2.1.1.2. In comparison to EDSN and Nodes, there is little information available regarding TW-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, a novel ADCS and a novel 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 time far longer than would have been possible with an uncontrolled orbit. TW-1 collected data on aircraft flight patterns using an on-board Automatic Dependent Surveillance (ADS) receiver. TW-1 also performed earth

observation by using visual spectrum cameras to image polar regions. In line with its objective, TW-1 also carried out a number of in-orbit tests on its experimentation sub-systems. S2G communication of the mission's various results was demonstrated at a rate of 125kbps, far beyond the rates achieved by Nodes. Apart from this figure, there is no further relevant information available regarding TW-1's energy or communication capabilities.

There are several other missions, besides EDSN, Nodes, and TW-1, that can provide insight into the state of the art of CSNs: 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

Inspiration for the design of CSN communication protocols can be drawn from several fields of terrestrial communications research. In this section two such fields are examined, both of which bear numerous similarities to the field of CSNs. These fields are Wireless Sensor Networks (WSNs) and Mobile Ad-Hoc Networks (MANETs). Both fields have considerable breadth and depth, as such a focus is placed on survey and review style publications. More focused publications are discussed in the subsequent "CubeSat Communications" section.

The following sections on WSNs and MANETs attempt to identify the most relevant sub-domains within each field. Within the field of WSNs works relating to data collection and energy conservation are of most interest. As discussed the primary application of CubeSat is sensing which leads to obvious similarities to terrestrial WSNs. MANET related works

are relevant in their treatment of the mobility of network members. Particular attention is paid to Flying Ad Hoc Networks (FANETs). FANETs share many of the same properties as CSNs. Like 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 WSNs challenges relate to the unpredictable or intermittent failure of network elements and resource and/or capability constraints. These challenges impact, to varying degrees, on a WSN's ability to perform data collection or data dissemination. The sink (collection) or originator (dissemination) of data in the context of CSNs is the earth based ground station. In this work we are concerned with energy efficient data collection (the PvTP trade-off) with the added complication of node (CubeSat) mobility.

A survey by Rault et al. focuses on energy efficiency in WSNs [66]. The authors approach the domain by examining a number of areas of WSN application such as healthcare, transportation and industry. For each area the authors outline characteristic WSN application requirements such as scalability, mobility, and security. Table 1 extends this table with a number of 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. ET Science examples: measuring solar radiation, performing astronomical measurements etc.

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

Rault et al. address several aspects of low-power WSNs and the trade-offs required when attempting to increase energy efficiency. Several low power WSN standards are explored such as IEEE 802.15.4 [67], Bluetooth Low Energy (BLE) [68], and 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 many extensions of 802.15.4.

Beyond low power WSN standards, several core WSN approaches are highlighted by Rault et al. Although all of these areas 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, in the Proposed Protocols chapter.

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 approaches, or cross layer optimization (CLO), and Multi-objective optimisation (MOO). Of the three techniques, CLO bears the most relevance to CSNs.
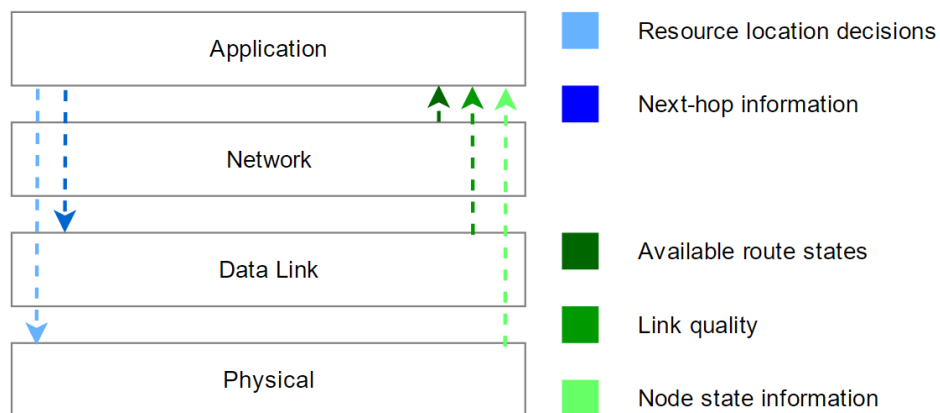
Figure 12. Examples of information that may be passed between layers. This is a departure for the treatment of surrounding layers as independent black boxes.

The authors define CLO as "solutions exploit(ing) interactions between different layers to optimise network performances" (Figure 12). Here 'layers' may refer to several communication models, but it can be safely taken as relating to the OSI reference model [25]. This work adopts Rault et al.'s definition of CLO from here on. Two particular surveys are cited as being authoritative on the WSN CLO domain [71, 72]. Rault et al. highlight the communication of radio energy consumption by the Physical layer to the MAC and network layers as an example of the application of CLO. Other examples given include: Communicating clustering formations achieved at the MAC layer and to higher layers and the explicit scheduling of communications at higher layers which allows lower layers to reduce activity.

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 field. CLO is a common and favoured theme throughout recent literature relating to WSNs and MANETs. The exact meaning of CLO differs somewhat depending on the author, application and domain. Despite this, it is clear that it is an active topic in both WSN and MANET energy efficiency research.

WSN data collection is examined by Francesco et al. in extensive 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. lay out a number of WSN mobility scenarios the most relevant of which is the "Mobile Peer" scenario (Figure 13). With CSNs, there is no appreciable difference whether one considers the ground station as moving into range of a satellite or a satellite

moving over ground. In either case, a network of satellites in some formation must contend with a finite but generally predictable communication window with ground.

Figure 13. 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 their 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. authors outline 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 schedule based discovery of reachable nodes. The timeliness of the discovery carried out by a mobile element determines the 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 attempts. Francesco et al. highlight an 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 learning are noted for their potential to avoid the pitfalls of periodic/scheduled listening. Such methods may, with or without prior heuristics or topology knowledge, converge on optimal discovery schedules over time [75]. In the case of CSNs, CubeSats can easily determine their next communication window with ground through the use of GPS time, positon and velocity. However, discovery of S2S windows closely mirrors the challenge of WSN mobile element discovery.

The data transfer phase, which follows the discovery phase, is primarily concerned with the quality of communication 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 approaches specifically adapted for WSNs.

Routing in WSN's, as highlighted by Francesco et al., is considerably more developed than discovery or data transfer. The authors make an assumption that the motion of network

elements is not controlled. In this regard, the works discussed are relevant to S2S communications of CubeSats which are assumed to have no orbital control capabilities. Routing for uncontrolled mobile WSN elements is classified by the authors into flat routing and proxy-based routing. In flat routing schemes all nodes behave in the same fashion whereas in proxy-based schemes certain nodes may take on routing or proxy style 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 passing velocity information between nodes. With this added information nodes can estimate the future stability of an inter-element link and avoid unnecessary abandonment or predict route switches. EARM implements an adaptive 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. Cluster heads act as routers to the mobile sink and manage inter-cluster communication. Somasundara et al.'s approach is similar to the cluster based MAC approach developed by this work. Although, this work does not employ a cluster based routing strategy such as the Cluster Based Routing Protocol (CBRP) [82] (Figure 14). 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 approach.
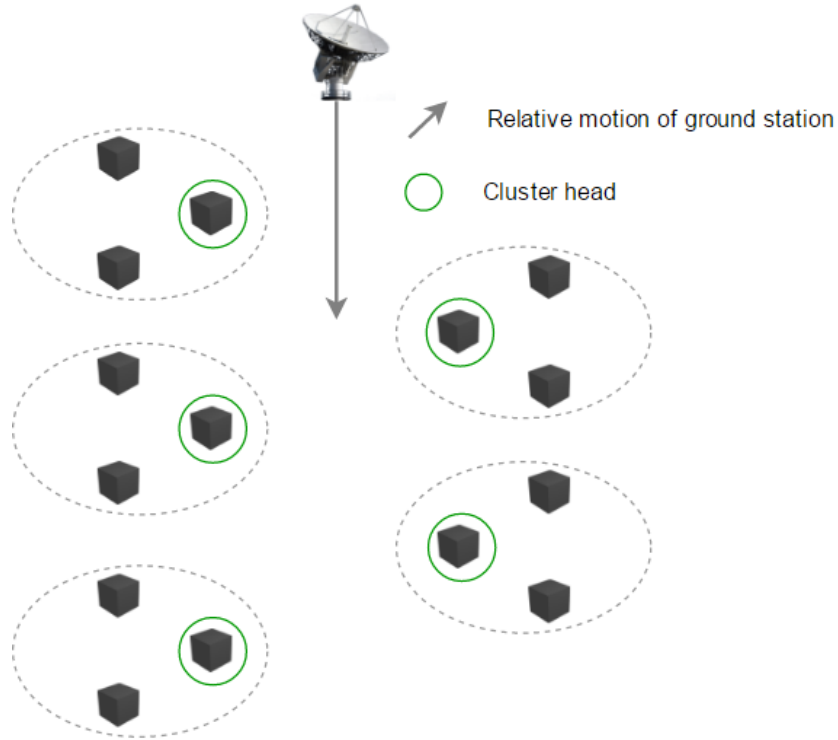
Figure 14. An example of network cluster forming with the election of cluster heads. With the introduction of a mobile sink (ground station), clustering can be adapted to insure election of cluster heads with longest contact duration and/or most resources [81].

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 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 determined the formal properties of MASP and used 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 heuristics.

The work of Gao et al. represents a state of the art approach to the PvTP trade-off in WSNs. Energy as a routing metric was clearly noted by Rault et al. as critical when attempting to optimize data collection with energy efficiency. MASP develops on this concept of energy as a routing metric and introduces a complex, yet performant, solution. However, MASP can only determine efficient routes given successful and timely discovery. Without the introduction of an appropriate discovery scheme MASPs performance is fundamentally limited. This lack of development in the area of discovery is further discussed by Francesco et al.

From the analysis of WSN based prior art one can identify several similarities to the CSN PvTP trade-off. In the area of energy efficient data collection, there are clear recommendations for further development of CLO and node discovery. Although the CSNs diverge in certain regards from WSNs, prior art therein provides a considerable basis for the development of the CSN domain.

2.2.2 Mobile Ad-Hoc Networks

MANETs bear obvious similarities to CSNs. MANET research tends to be less concerned with the constraints, such as power and bandwidth, placed upon network members. Rather, a focus is placed on an approach's ability to efficiently and reliability enable self-organization and communication.

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 are those which are mature, adaptive to topology changes and energy efficient.

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, generally 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 differing 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 where the additional overheads of more complex protocols scale more favourably.

Several routing 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. Non-target nodes add their address to an incoming RREQ packet and rebroadcast it. Loops are avoided by nodes dropping RREQ packets to which they have already added themselves. Once an RREQ reaches its target it contains a list of nodes representing one possible route from originator to source. 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 is capable of building up multiple routes for a given target. In DSR, due to the construction of routes within packets, packet sizes grow in proportion to 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 specifies that the originator and target addresses must be contained within a route packet. AODV borrows route packet sequences numbers from DSDV in order to avoid infinite loops and determine the 'freshness' of a route. 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 has a route to for a given target. 'Hello' messages may be periodically generated in order to detect broken routes and maintain up to date route costs.
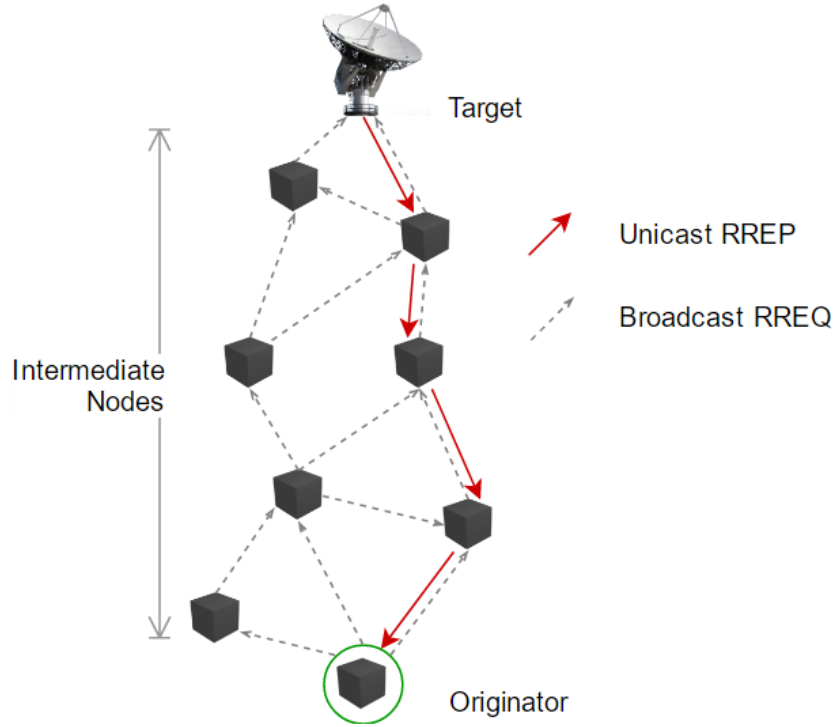
Figure 15. An illustration of the broadcast RREQ unicast RREP approaches used by DSR and AODV. Targets will generate RREPs for each arriving RREQ. RREQs which revisit nodes, and are dropped as a result, are not shown.

The routing protocol utilized by this work, DYMO, is a modification of AODV. In fact, 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 Proposed Protocols chapter.

Vehicular Ad-hoc Networks (VANETs) are a sub-domain of MANETs. There are notable parallels between CSNs and VANETs such as the separation of vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) concerns. 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, secure communication 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 stack. Implementing a stack which supports IPv6 enabled RPL [69] for CSN communications is potentially feasible. However, in broad terms, the power use, overheads and low mobility basis of 802.11 flavours are generally unfavourable. Mohammad et al. point to C2CNet (Car-to-Car Net) as the emerging state of the art standard within 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 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 antennae 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]. Interestingly, due to the use of time slots for

route discovery, the aforementioned modified AODV protocol has similar properties to the combination of protocols proposed by this work.

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. 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 FANETs further reinforces its importance to the future of CSNs. Bekmezci et al. note works that take advance of cross-layer clustering and scheduling and 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. It is clear that the FANET sub-domain contains the highest relevancy work. Overall, examining MANET prior art augments and reinforces the findings relating to WSNs. Such prior art provides important context to the existing state of 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 publications examined the inter-communication and networking of CubeSats. Most notably Challa and McNair of the University of Florida provide explorations of distributed applications implemented upon CSNs [37, 99-102]. These works are out of the scope of this project as they deal more with applications running upon CSNs rather than communications within the CSN itself.

The most relevant work in this area is a survey by Radhakrishnan et al. [9]. This survey provides the primary source for the exploration of the CubeSat communications state of the art. This work proposes MAC (data link layer) and routing (network layer) protocols to address the PvTP trade-off. Radhakrishnan et al. detail several relevant works relating to MAC, routing and energy efficiency.

Radhakrishnan et al. provide an overview of some of the common terms used when referring to CSN formations. A 'trailing' formation, sometimes referred to as 'leader-follower', involves a single orbit chain. A 'cluster' formation has a broader mean. "A cluster of satellites" generally implies a collection of satellites in multiple orbits which maintain some topology or formation. The term 'constellation' is often misused. A constellation formation focuses on coverage of the earth's surface. Communications and GNSS constellations typically seek to achieve complete coverage or 'visibility' of the earth's surface. The term 'swarm', is also often misused. A swarm is not a satellite formation in the same sense as a cluster or constellation. To quote 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. Similar to the

concept 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 16). In this work we consider CSNs which fly in cluster formations and operate as swarms.
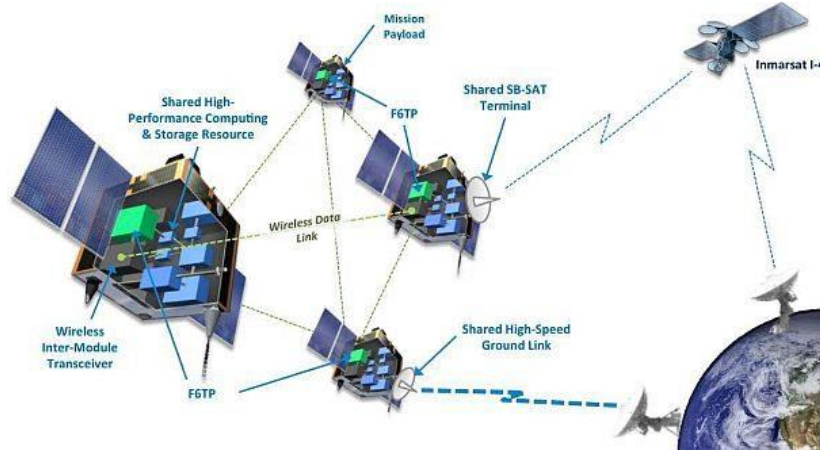
Figure 16. DARPA fractionate satellite concept. It can be seen that the mission payload exists independently of other core systems such as compute and communications. Image Credit: DARPA

2.3.1 Physical Layer

Although the physical layer of OSI model is not of core interest in this work it is nonetheless worth noting some of the findings of Radhakrishnan et al. When referring to the maximizing data rates Radhakrishnan et al. recommend the increase of 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]. This intermittent S2G communication approach is adopted by this work. 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.

In comparison to other areas of CubeSat communications, there has been notably more work published relating to antenna design. Radhakrishnan et al. point to Gamalink [42] for its use of an advanced antenna. Unfortunately, no further information regarding Gamalink is offered. Single patch S-band (2 – 4 GHz) antennae are highlighted as the current state of the art. Also, the authors state that "a maximum 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 the work previously discussed in relation to CubeSat communication capabilities. Radhakrishnan et al. express doubts regarding the applicability of complex MIMO and multi-patch antenna, instead recommending the use of multiple simple antennae. Radhakrishnan et al. state that links between satellites are typically full duplex typically favouring Time Division Duplex (TDD) over Frequency Division Duplex (FDD).

2.3.2 Data Link Layer

Thus far the data link, or simply 'link', layer has been referenced in relation to Medium Access Control (MAC). Although the focus in this work will continue to be placed on MAC protocols, it is worth noting some of the other duties of the link layer such as: framing, synchronization, error control, flow control and MAC addressing. MAC protocols determine how multiple agents share a common radio medium. As such, 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) rely on detecting when the medium is in use and when two

signals have collided on the medium (collision). Contention free protocols seek to completely avoid the need to detect medium use or collisions. Such protocols generally achieve this by allowing multiple agents to communicate at once without collision such as with Code Division Multiple Access (CDMA) or by logically dividing the medium such as with Time Division Multiple Access (TDMA) or Frequency Division Multiple Access (FDMA) (Figure 17).
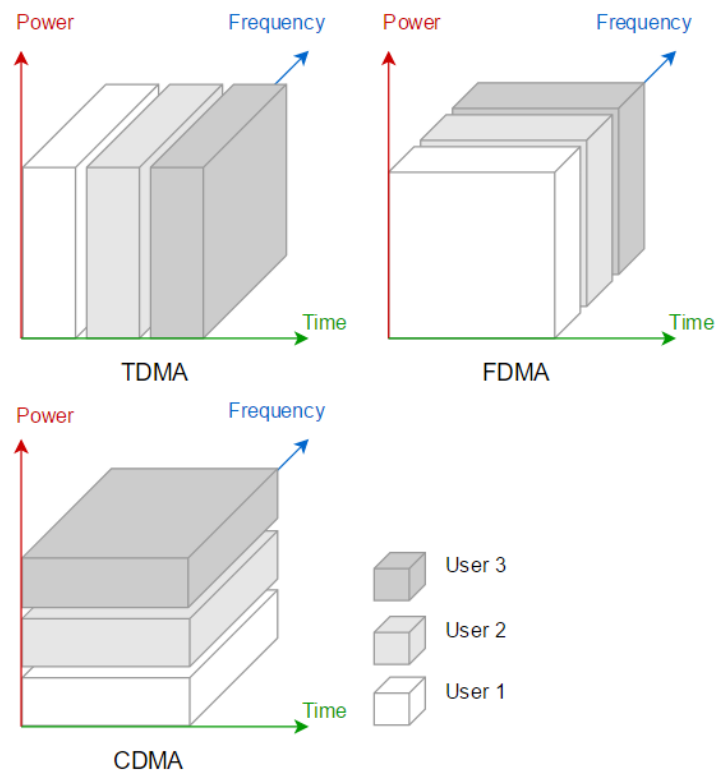
Figure 17. A comparison of common contention free MAC schemes. In CDMA a 'chip' is used to insure that signals on the medium are orthogonal and therefore cannot collide.

Several MAC protocols are discussed by Radhakrishnan et al. in relation to small satellite communication. One approach attempting to adjust the IEEE 802.11 physical and MAC standard for communications in LEO [106]. As discussed in relation to VANETs several aspects of 802.11 based standards are unsuitable for space based 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 modified 802.11 approach adjusted contention windows to fit communication delays. The distributed adjustment of IFSs and contention windows introduces considerable complexity at scale. The designers of this modified 802.11 approach assert its feasibility for use in LEO communications. However, it is this Author's opinion that the modifications introduced solve 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 see the medium as 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. In their assessment of this CSMA/CA approach Radhakrishnan et al. conclude that the protocol is best applied in situations with low frequency communications within tightly grouped formations. Generally, CSMA and other contention based schemes are liable to introduce unnecessary and wasteful communications adversely affecting power consumption.

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 for the 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 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 TDMA 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 17). Each time slot within a schedule is typically assigned an owner. During a time slot only the slot owner may transmit. There are many different flavours of TDMA. For instance schedule 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. Nodes in LDMA may be in differing modes at any one time. As such, when a node is in LCL mode it 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 from LCL to HCL based on the number of conflict frames received. Nodes will revert to LCL mode as their communication demand drops and the number of overhead conflict frames reduces.
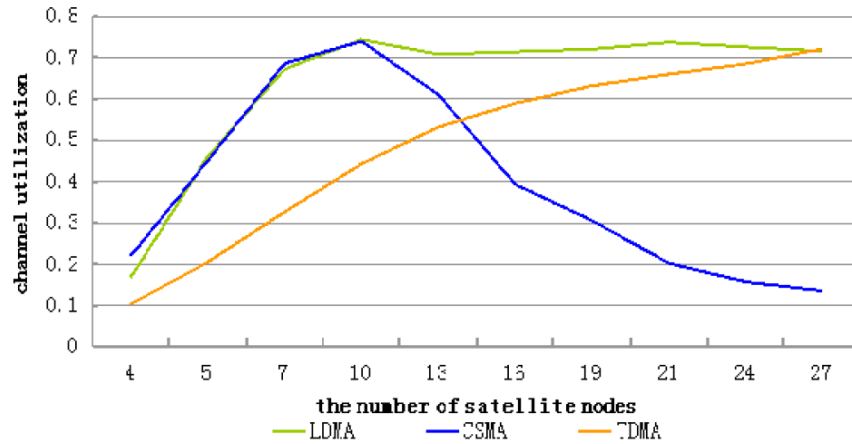
Figure 18. Cannel utilization (Vertical axis), measured from 1.0 (100%) to 0, compared to network size (Horizontal axis) for LDMA, pure CSMA and pure TDMA. Image Credit: [109]

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 heavy loading 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. compared LDMA to pure TDMA and CSMA protocols through simulation. One graph representing their findings is shown in Figure 18.

LDMA presents itself as strong candidate for use in attempting to optimize throughput for CSNs. As communication activity falls the protocol mirrors a pure CSMA approach and it follows that the protocol approaches pure TDMA as activity increases. The power 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 a power consumption

52

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 discusses by Radhakrishnan et al.: An FDMA/TDMA (F/TDMA) hybrid and a CDMA/TDMA (C/TDMA) hybrid. The F/TDMA protocol, based on WiMedia [110] and proposed by Heidari et al. is worth mentioning briefly. The protocol introduces the distributed management of heterogeneous network state as well as two-dimensional super frames in place of TDMA time slots. The C/TDMA protocol takes a cluster based approach to the division of CDMA and TDMA [111]. Comparatively the C/TDMA protocol provides many of the same properties as the aforementioned F/TDMA protocol without the prohibitive levels of complexity.

The C/TDMA protocol was chosen as the starting point for the MAC protocol proposed by this work. As such, the details of the protocol will be discussed at length in the Proposed Protocols chapter rather than here. The protocol was chosen as it makes explicit allowances for energy awareness and the improvement of throughput. For a given scenario involving a trailing formation, simulations of the protocol show C/TDMA out-performing the previously discussed CSMA/CA approach (Figure 19). 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 other nodes within the cluster to conserve energy.
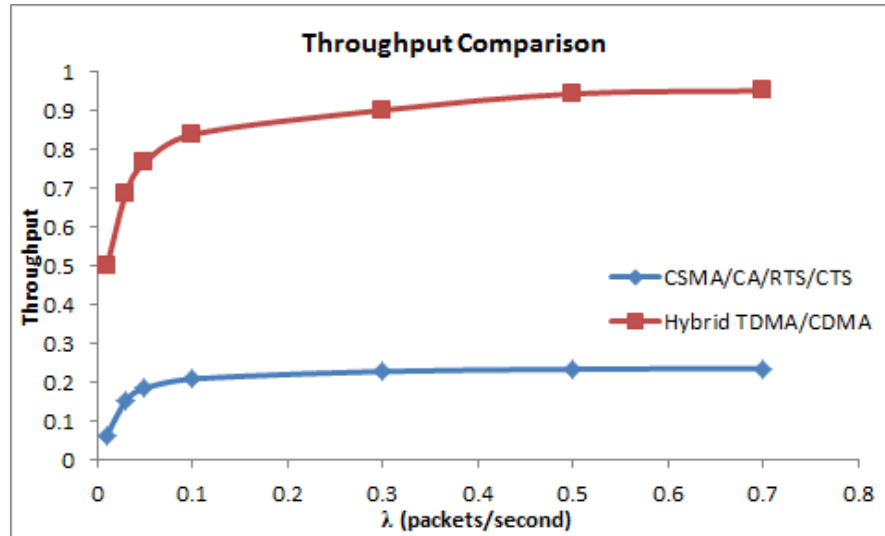
Figure 19. Throughput here is measured as the amount of time spent transmitting data divided by the amount of time available to transmit data. Considering the findings in Figure 18 It is likely that the experimental setup and the traffic simulated naturally favoured the T/CDMA protocol. Image Credit: [111]

2.3.3 Network Layer

The primary responsibility of the network layer is routing, although packet forwarding and address handling are also important activities. Routing protocols affect the discovery and selection of optimal routes within a network. There are several generally applicable approaches to the discovery of routes 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 nodes with a low amount of remaining power. The approaches to determining and maintaining optimal routes differentiate many of the routing protocols proposed for small satellites. It should be noted that there has been less published work relating to the routing protocols than to MAC protocols. As evident from the Nodes and Tianwang missions, the current state of the art for CubeSats 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 [9]. 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]. Unfortunately, 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 but no notable works on this have yet to be produced.

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 reduces the overall frequency of route discovery. The protocol mainly focuses on synchronization of larger satellites with regular control orbits. However, 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 of 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 that further work be carried out on routing is provided however, there is clear need for such work. 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 network layer protocols. 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 space to ground communications be performed indirectly through relay with existing space bound communication networks [10]. This concept is further explored for deep space missions in the preliminary development of the ESA's COPINS mission [60].

Another survey style paper on inter-satellite link for CubeSats by Budianu et al. [8] published in 2013 provides a somewhat out of date overview of the field. However, Budianu et al. pay more attention to antenna design and link budget analysis than Radhakrishnan et al.

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 illustrate further the considerable context which must be considered when approaching issues relating to CubeSat networking and communications.

2.4.1 Energy Aware Scheduling

Energy aware scheduling is an active area of research in the terrestrial domain, especially in relation to WSNs [115]. Despite the growing popularity of small form factor satellite missions, there are few related notable publications in the domain. However, new insights into the area was produced in 2016 as a result of the technology demonstration focused GomX-3 mission [116]. GomX-3 was designed by private Danish company GOMSpace and flown by ESA. Following the mission's success mission designer published a work entitled "Battery-Aware Scheduling in Low Orbit The GomX-3 Case" [61]. The work outlines the approach taken to adaptively model and predict battery usage in order to produce a activity schedule which optimized power consumption. In relation to long duration CSNs, one might consider an energy conservation scheme which entails generating an optimal schedule for each CubeSat. Through distributed consensus these schedules could then be used in order to generate final desired activity schedules.

2.4.2 Delay Tolerant Networking
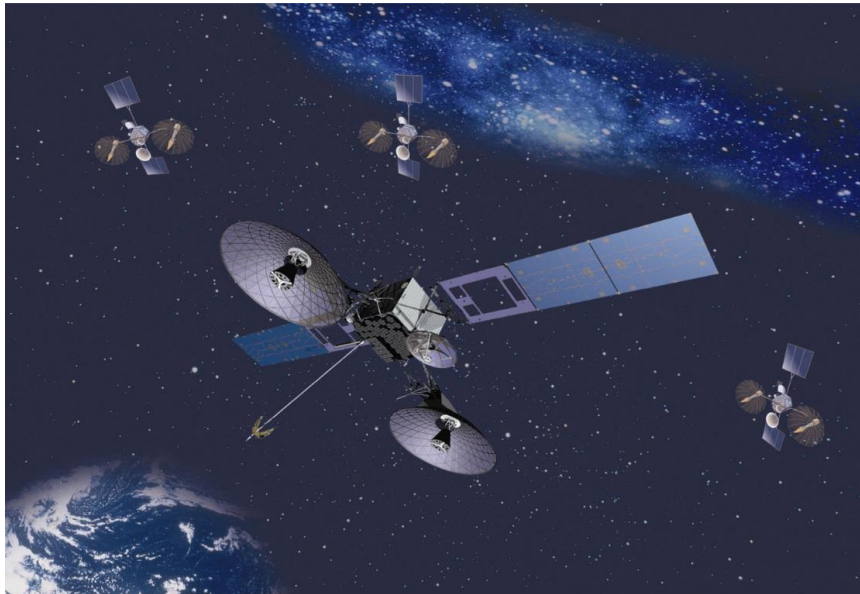
Figure 20. An illustration of several NASA Tracking and Data Relay Satellites (TDRS) which make the primary backbone of NASA's deep space network. Some of the most advanced space bound radios enable the relay of signals from spacecraft throughout the solar system. Image Credit: NASA

Interplanetary communication presents many novel problems. Delay tolerant networking (DTN) approaches have been employed successfully solved many of these [117]. A notable success case is NASA's deep space network [118] (Figure 20). 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 obvious advantage in this regard of being tested on-board larger satellites and developed by experts within the space industry.

Chapter 3: Proposed Protocols

One-hop network topologies are the current state of the art for CSNs. In such networks, once a node has discovered its neighbours there is no need to establish routes. For example, consider EDSN's "hub-and-spoke" topology and Cpt/Lt protocol (Figure 3). If EDSN CubeSats are not in range of one another they cannot communicate. Given this relationship, the performance characteristics of overall S2S communication is primarily determined by aspects of the physical and data link layers. Section 2.3.1 briefly discusses the state of the art of CubeSat physical layer implementations, as assessed by Radhakrishnan et al. This work's simulations are configured with idealized physical layers, the behaviour of which is considered immaterial to gathered results. Addressing and modifying such implementations is beyond the scope of this work.

This work is considered with data link layer MAC protocols. Although, as discussed in section 2.3.2, there are other aspects of the data link layer which may affect the PvTP trade-off. In any wireless network the method by which the medium is shared by nodes will impact various aspects of performance. Approaches such as TDMA and CSMA are commonly used for simple one-hop networks. Even in simple one-hop networks there are numerous factors to consider. If nodes are closely grouped, with each node having multiple neighbours, then TDMA is likely to outperform CSMA due to its avoidance of collisions. However, if nodes in a closely grouped network only sporadically generate traffic and contention is low then CSMA may outperform TDMA. The optimal choice of MAC protocol is rarely obvious and is affected by further factors such as the network's application, topology, physical layer capabilities, node mobility, and environment.

Two MAC protocols from the reviewed literature were chosen as potential candidates for the basis of this work's proposed MAC protocol: LDMA [109] and C/TDMA [111]. Ultimately, C/TDMA was chosen. LDMA offers the potential for the best aspects of both CSMA and TDMA. C/TDMA on the other hand, has the ability to selectively reduce the energy consumption of certain nodes through cluster formation. The final choice was motivated primarily by the greater availability of information for and simplicity of C/TDMA. As routing protocols are also of interest in this work, the estimated time required to implement C/TDMA was of significant importance.

This work makes a several changes to C/TDMA as specified by Radhakrishnan et al. These changes primarily relate to the enabling nodes to periodically conserve energy in response to network activity. 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.

As discussed, due to the current CSN state-of-the-art, routing protocols are currently of less significance to the PvTP trade-off. Despite this, it is clear from the development of Gamalink [42] and the recommendations of several domain experts [9, 12, 58], that multi-hop networks are the next stage of development for CSN applications. As such, this work proposes the use of DYMO [83] as a routing protocol for use with CSNs. DYMO is introduced briefly in section 2.2.2. This choice was primarily motivated by the availability of an existing implementation of DYMO for OMNeT++. Without this implementation it was unlikely that a suitable routing protocol could have been fully addressed by this work. As with C/TDMA, several changes are made to DYMO. These modifications are made mainly to resolve issues with the existing OMNeT++ implementation. This work's implementation of DYMO will be referred to as D3 (DYMO Cubed).

This work places the majority of its focus on CubeMac. Compared to CubeMac, the customization and the choice of basis for D3 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 protocol such as DYMO this aspect of CSN's would have required an idealized simulation approach reducing the final fidelity of results. Ultimately, both D3 and CubeMac combine through elements of cross-layer optimization to form the primary contribution of this work.

3.1 CubeMac

The proposed CubeMac protocol builds upon the work of Radhakrishnan et al.'s cluster based C/TDMA protocol. CubeMac is designed with direct consideration to the PvTP trade-off and the chosen hypothetical mission. CubeMac focuses on data collection and attempts to avoid distributed decision making where possible in an attempt to reduce energy consumption. Although CubeMac makes several potentially beneficial additions to the work of Radhakrishnan et al., it is clear that the protocol retains a number of weaknesses. The following sections detail the operation of CubeMac and discusses its potential strengths and weaknesses.

3.1.1 TDMA

TDMA, as discussed 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. For convenience this static approach will be referred to as 'pure' TDMA. Although the avoidance of more adaptive approaches is not discussed directly by C/TDMA's designers, pure 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 in such environments, although not impossible, are likely to be incur potentially unacceptable time and energy costs. Although pure TDMA may be outperformed in certain regards by various adaptive schemes, it avoids the need for such decisions making approaches. The Cpt/Lt protocol, although adaptive in its assignment of roles, uses a pure TDMA approach. The schedule and duration of time slots and their respective owners is fixed and known to all network members.

3.1.2 Cluster Formation

Pure TDMA is coupled with CDMA to form Radhakrishnan et al.'s C/TDMA. This integration is made possible by the introduction of clustering. Using a "centrality

algorithm", which is not specified further by Radhakrishnan et al., a CSN is divided into a number of clusters as in (Figure 14). Within each cluster a "master", similar in function to a cluster head, is selected. The remaining nodes in the cluster assume the role of "slaves". Inter-cluster communication is handled solely by cluster masters (Figure 21). Master-to-master and master-to-slave communication takes place using pure TDMA. Slave-to-slave and slave-to-master communications on the other hand uses CDMA. In short, a special time slot is dedicated to for the use of CDMA allowing all cluster slaves to simultaneously communicate with their cluster master. This behaviour is discussed further in the following section.
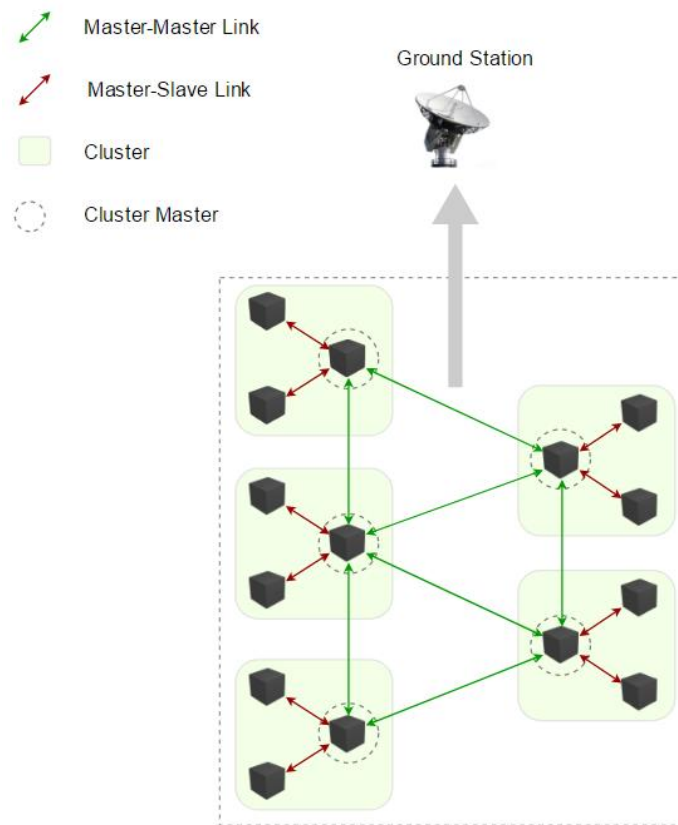
Figure 21. Inter and intra cluster communication in CubeMac. Note that slaves can only communicate with one another via one or more cluster masters.

63

It is worth noting that Radhakrishnan et al. specify two distinct types of TDMA frames, referred to by the authors as "channels". A frame in TDMA nomenclature is a group of time slots which follows some fixed repeatable pattern (Figure 22). In the specification of C/TDMA one frame is dedicated to 'uplink' and the other to 'downlink'. The use of the term 'downlink' here should not be confused with S2G communication which is often referred to as 'downlinking'. During uplink, frames slots are dedicated to allow slaves transmit to their cluster master and cluster masters to transmit to one another. During the downlink frame cluster masters focus on transmitting to slaves within their clusters.
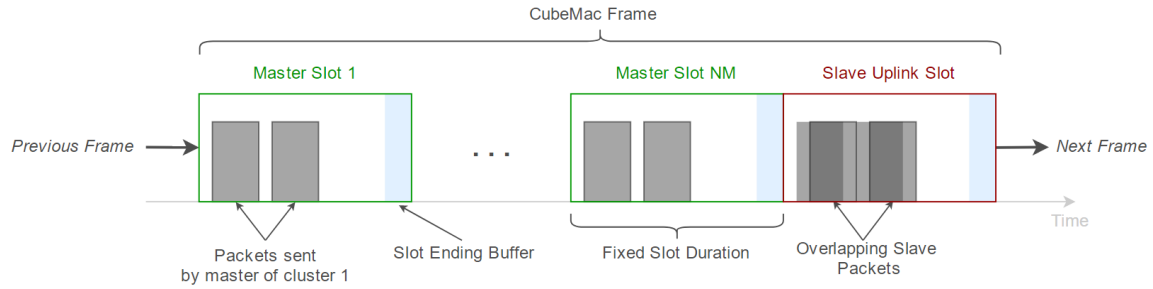
Figure 22. One complete CubeMac frame. "Master Slots" are collision free through TDMA. "Slave Uplink Slots" are collision free through the use of CDMA. Frames repeat indefinitely.

As discussed in section 2.2.1 this work is only concerned with data collection and not data dissemination. As such, CubeMac only implements a modified version of the Radhakrishnan et al.'s uplink frame as packets destined for ground will never be routed to slaves (Figure 23). This is a result of the behaviour of which manages the selection of which nodes may perform S2G communication. In this work's simulated 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 a cluster master. If this master is not a GM, these packets will be forwarded to through neighbouring masters

until reaching a GM (Figure 23). CubeMac's operation is unaffected by D3's behaviour, D3 is simply aware of the CubeMac's master and slave roles.
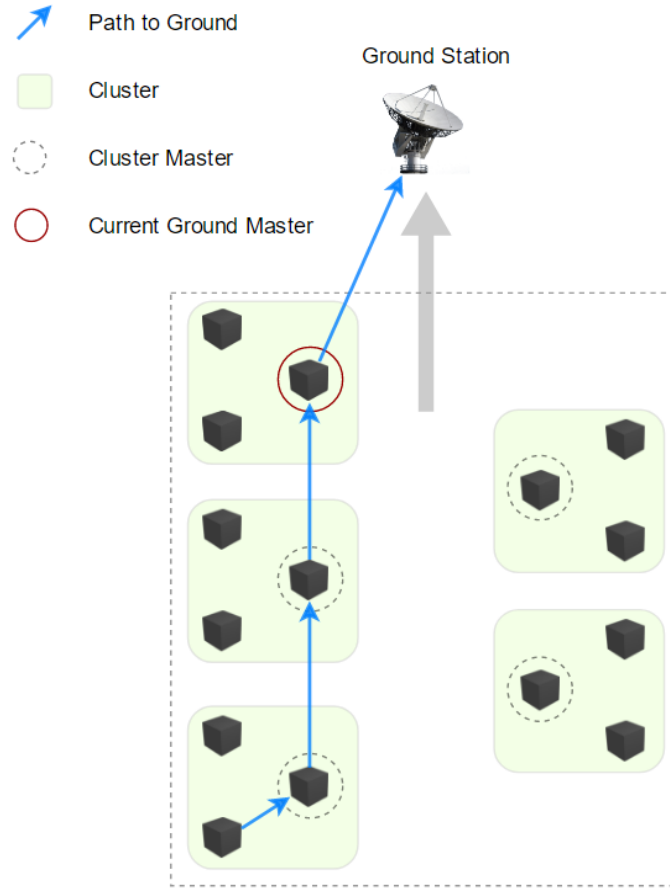
Figure 23. All valid paths from slaves to ground require one or more masters. The Ground Master always constitutes the final hop on a path to ground.

3.1.3 CDMA

Simulations of CubeMac take a highly simplified approach to the treatment of CDMA. In simulations only the additional energy costs and parallel communication of CDMA are modelled. This approach was deemed sufficient for the purposes of the exploring

CubeMac's effects on the PvTP trade-off. Nonetheless, it is worth briefly discussing the general operation of CDMA.

Figure 24. **FIX XNOR** A CDMA approach to spreading an initial message over a greater bandwidth using a given code. The original message contains four bits however the final signal is 24bits in length due to the 24bit code used. The XNOR operation will produce a '1' whenever both inputs are '0' or '1' and a '0' otherwise.

CDMA based protocols combine 'chip', or 'code', with messages in order to form a final signal which is spread across a greater bandwidth (Figure 24). The primary benefit is that multiple signals may share a wireless medium at once 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.

Parallel communication through CDMA comes at the cost of added complexity and overheads. As shown in Figure 24, spread signals require greater 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 often result in a several factor increase in energy consumption. CDMA also generally requires more complex and costly 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. Transmitters using CDMA incur a higher energy cost which is proportional to the amount of time taken to transmit the unencoded message. This simplification was motivated largely by the absence of an existing suitable CDMA implementation for OMNeT++.

3.1.4 Frame Structure

As alluded to previously, 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 will own the time slot corresponding to its cluster ID. For example, the master of the cluster with ID 1 (M1) will occupy time slot 1. The final time slot (N_m +1), named the "uplink" slot, is reserved for cluster slaves. During this slave uplink slot all slaves with each cluster use CDMA to transmit data to their cluster master. 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 transmitting, listening and sleeping states.

Figure 25 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-communication between all nodes within a network.

The behaviour illustrated in Figure 25 represents a slight departure from Radhakrishnan et al.'s 'uplink' frame. Cluster master 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 would instead reserve a separate 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 which no transmissions can occur from masters to slaves. In this work, such transmission are required by D3 for route discovery.

Figure 22 and Figure 25 illustrate that each time slot ends with a short buffer period. This buffer is included to combat clock drift across the CSN. EDSN mission designer estimated the mission's maximum clock drift as 12ms. The buffer also acts to alleviate the effects of jitter. Nodes must take into account this buffer when predicting how many packets to attempt to transmit within any given slot.

3.1.5 Energy Conservation

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 25, 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 slot owned by the cluster masters, it is a simple process for slave to determine slots in which they may sleep.

The second feature allows nodes to sleep during slots under various conditions. In implementing this feature CubeMac introduces a special field with packet headers. This field, the "last" field, allows nodes to indicate that a packet is the last they intend to send during a slot (Figure 26). CubeMac also introduces a timeout period, which is not specified in C/TDMA (Figure 26).

CubeMac requires each cluster master to track the number slaves present in its cluster. This is achieved by the addition of a packet header field which contains the ID of the cluster to which the packet originator belongs. Finally, CubeMac introduces the concept of an optional "no data" (ND) packet and an ND header field (Figure 27). A node, without packets to send, may broadcast an ND packet to those listening to indicate that it is has no packets to send during this slot. ND packets are made as small as possible and are important in cases where multiple nodes are waiting in listening mode. Using these additions and the general structure of CubeMac, there are several conditions under which a node may sleep during a slot.

In the master role:

1. If 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. If, during a non-uplink slot, the slot owner sends an ND packet or a packet marked as its last

3. If, during an uplink, all slaves transmit an ND packet or a packet marked as their last

4. Following the transmission of an ND packet or a packet marked as "last"

In the slave role:

1. If no packet is received from the cluster master during the cluster master's slot for a period longer than the configured timeout period

2. If, during the cluster master's slot, the cluster master transmits an ND packet or a packet marked as their last

3. Following the transmission of a packet marked as "last" or an ND packet
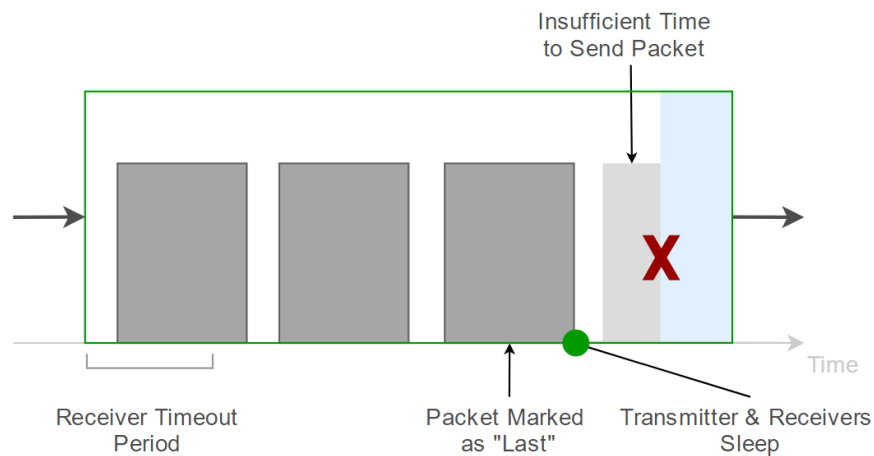
Figure 26. The last packet transmitted by any node within a slot will contain a flag indicating that no further packets should be expected. In slave uplink slots masters must wait to for all slaves in the cluster to send a "last" packet.
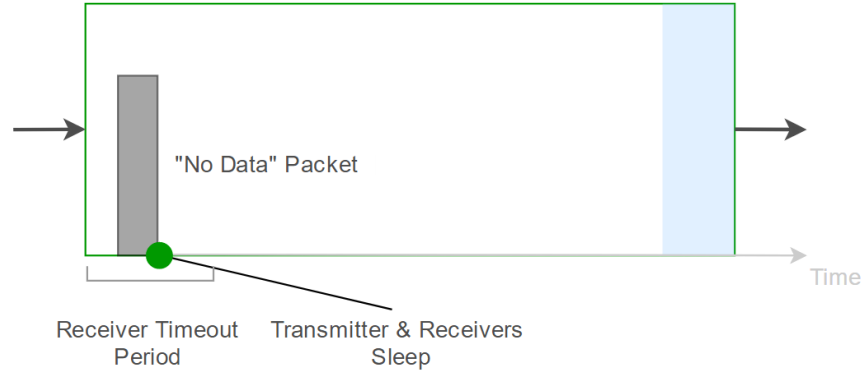
Figure 27. In order to allow multiple nodes in receiver mode to sleep, node will generate a "No Data" packet when they have no data to send during their slot. Sending this packet incurs an energy penalty but this out-weighed by allowing multiple nodes to sleep prior to a timeout.

These added energy saving features incur a low cost as a result of C/TDMA's static TDMA approach and cluster based architecture. The only tangible overheads introduced are those of the added packet header fields. The "last" and ND fields only require one additional bit and the cluster ID field only requires $\log_2(N_m)$ bits. Crucially these energy saving features do not come at the cost of throughput. These features take advantage of the correspondence between slots and cluster IDs as well as the master-slave relationship with the reducing available windows of communcation.

3.1.6 Drawbacks

There are two notable drawbacks of the added energy conservation features to consider. The first relates to the master-slave relationship. The restriction of slave communications to cluster masters is fundamental to CubeMac. To lift this restriction would require introducing addition slots for each slave within a cluster at which point CubeMac essentially approximates pure 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. The second major drawback is one inherent in effectively

71

all TDMA or schedule based systems. If a new packet is generated the moment after a node makes the decision to go to sleep it will have to wait for an entire frame before being sent (*Poorly timed packet*). This is fundamentally the same issue as occurs with packets generated immediately following the end of a node's slot. Once again however, this drawback is of little concern in the context of this work. Ill-timed packet generation will affect latency but overall will not affect throughput. In this regard CubeMac sacrifices on latency in order to reduce energy consumption.
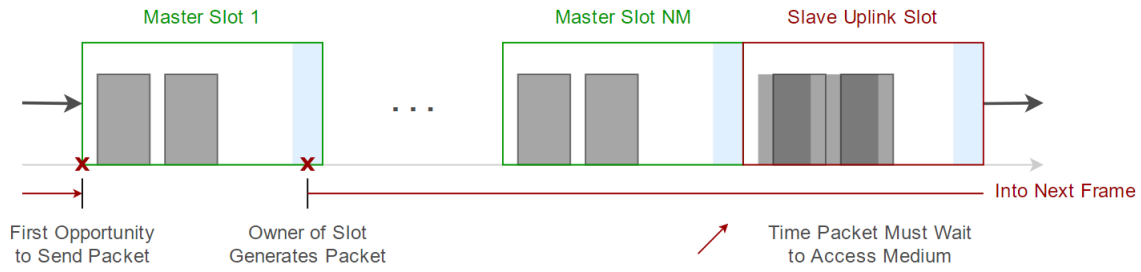
Figure 28. Packets generated within a the buffer period or directly following the end of a slot must wait a minimum of approximately N_m slot durations before transmission.

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 in a time slot then the medium is being poorly utilized. In CubeMac this master could not borrow time from another time slot in order to transmit its additional packets despite the fact that all other time slots are 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 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 NEEDS WORK The use of global time slots also creates an issue in relation to CubeMac. It is inevitable that nodes within CSN will be widely spatially distributed. If 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 result of global time slots. In cases where time slots are dynamically allocated at a local scope 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 some basic carrier sense based approaches during the assignment stage to overcome issues inherent with local time slot assignment such as hidden node problem discussed in section 2.3.2.

The general nature of CubeMac makes it ill-suited for node failures. For instance, if a master's 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 21). 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 selection are important in addressing challenges related to node mobility and failure. Neither feature is implemented in this work as the PvTP trade-off is examined during "steady-state" network operation. It is worth noting that

73

both 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. Otherwise, master nodes would quickly exhaust their available energy supplies and all slaves would become isolated from ground. Radhakrishnan et al. mention master selection and clustering briefly but provide no tangible specification of either. This work examines CubeMac during steady state operation following clustering and master selection. Both dynamic clustering and master selection are also discussed in the context of future work in section 6.2.

3.2 D3

D3 does not diverge significantly from that of DYMO the behaviour of which is highly similar to that of 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 15). 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 was 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 29). This RREP will be identical to an RREP a target receiving the same RREQ would generate.
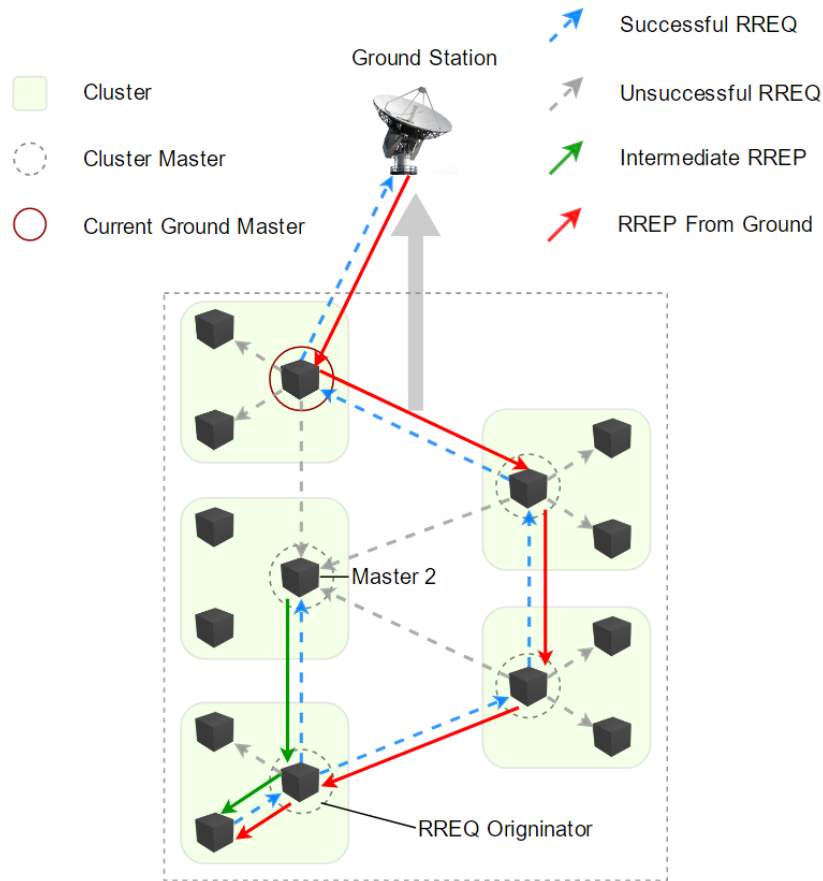
Figure 29. Master 2 generates an intermediate RREP as it already has a route to ground. The original RREQ for this intermediate RREP had to travel two hops as opposed to the minimal four hops to ground. The RREP from ground may replace the route generated from the intermediate RREP if its route has a 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 be the cluster master. If the cluster master has a route for a target it can generate an RREP thus reducing 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 ground. 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 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 routes held by intermediate nodes are used.

3.2.2 RERRs
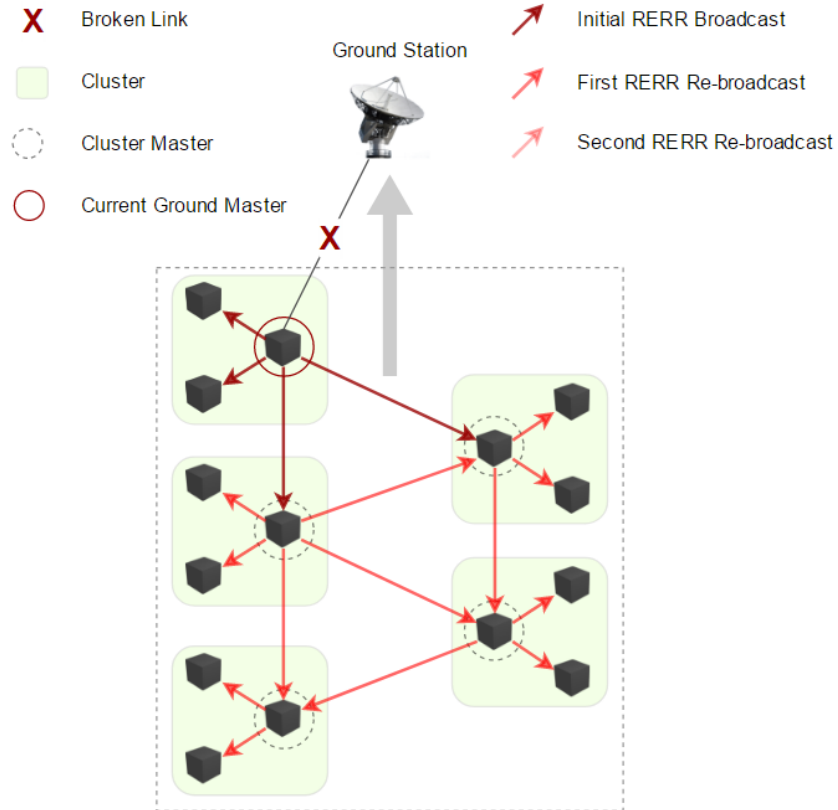
Figure 30. The broadcast based propagation of a RERR message throughout the CSN. In this case, all nodes receiving the RERR will discard their routes for ground.

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 in this work of link breakages. As will be discussion further in section 4.2.2, master nodes use a separate network interface for S2G communication. This interface uses a MAC protocol based on CSMA rather than CubeMac. Unlike CubeMac, this CSMA protocol does utilize acknowledgements. As ground masters move out of range of ground this CSMA protocol signals DYMO that the 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 30.

3.2.3 Sequence Numbers

Like AODV, DYMO utilizes 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 and as such 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
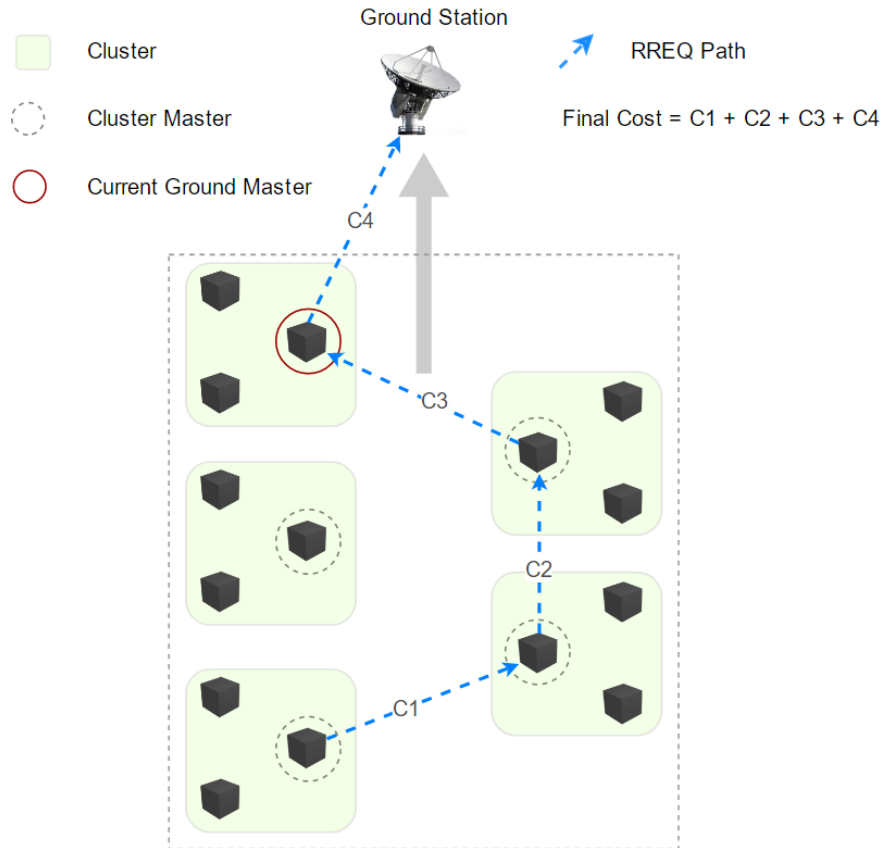
Figure 31. As an RREQ, or RREP, packet moves through a network it accumulates link costs. Costs are added to a cost field within the packet upon receipt. Accumulated costs are used within route entries and for the detection of routing loops.

As previously mentioned, DYMO allows node to record each route's 'cost'. These costs are central to DYMO's approach to routing loop detection. Route costs are established within RREP packets. As an RREP is transmitted unicast from the target node to the originator node, each intermediate node adds some greater-than-zero value to the RREP's

'cost' field (*Link Costs*). As the name suggests 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 (*Link Costs*).

The most common and basic cost value is that of "hop count". When using a hop count cost function nodes simple 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 battery capacity < N add '10' to link cost – else add '1' to link cost. Provided all node's use the same cost logic, originators will favour routes which avoid nodes with battery capacities below the threshold of N. If all nodes for all possible routes have battery capacities above this threshold then the shortest route 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. All of these approaches represent some degree of cross layer optimization, a common topic discussed in chapter 2. 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 cost of announcing a 'cheap' link cost.
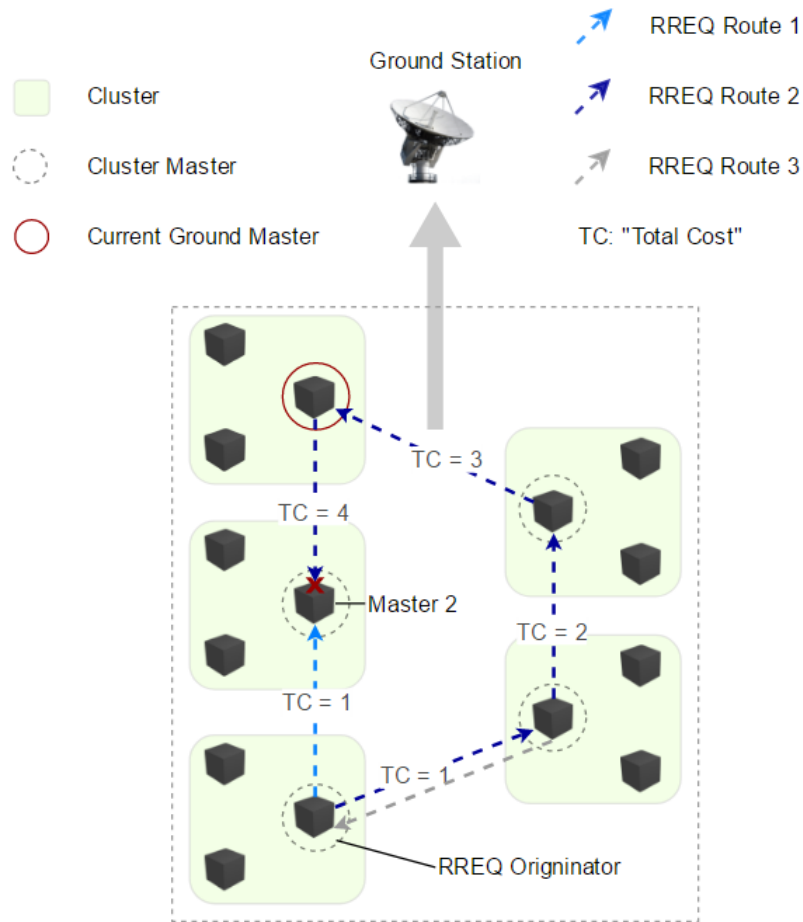
Figure 32. DYMO detects loops by compared a route messages cost metric with that of a corresponding route for that route message. Following RREQ route 1, Master 2 will generate a route to the Originator with a cost of '1' (Hop Count). Master 2 will drop the RREQ taking route 2 as it already has a lower cost route to the Originator. RREQ route 3 will be dropped by the Originator as the RREQ's originator field matches its own address.

DYMO is capable of detecting routing loops using route costs as a result of its focus on opportunistically establishing routes. Unlike some other routing protocol, DYMO route entries only require a target address and a next hop address. For instance, when a node X receives an RREQ from another node Y and the RREQ has an originator of Z, it follows that a route is available to Z and that Y is the next hop (Figure 32). When X first sees an RREQ it will add this route to Z and then broadcast the RREQ. Y will see this RREQ from

X, however, *Y* has enough information to avoid rebroadcasting this RREQ and causing a broadcast loop. *Y* already has a route to *Z* and the RREQ from *X* will have a higher cost as all added cost value must be greater than zero. This allows *Y* to drop the RREQ. This approach avoids routing loops and the same logic can be used to avoid loops of arbitrary lengths (Figure 32).

3.2.5 Discovery and Maintenance Patterns

Through the use of route messages, sequence numbers and route costs, DYMO has suitable capabilities to discovery, chose and detect the breakage of routes. The remaining salient aspects of DYMO relate to its timing of route discovery and maintenance. By default DYMO includes no explicit route maintenance. Like AODV, DYMO has a specification for the "Hello" messages which are intended to help maintain up to date routes and route costs. Such messages are introduced in section 2.2.2.
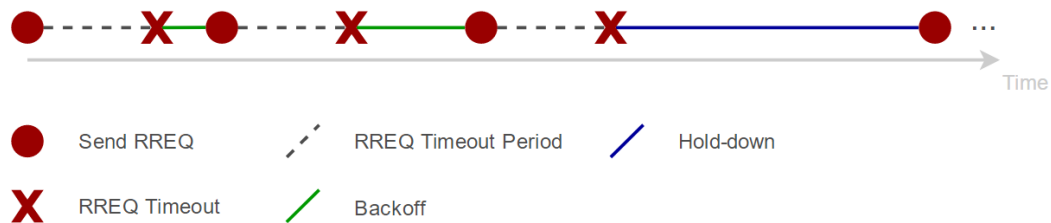
Figure 33. A node attempts three route discoveries by broadcasting an RREQ. In each case the RREQ timeout period elapses before an RREP is received. Following a time a back-off period is observed before sending another RREQ. Failing the maximum number of sequential discovery attempts send the node into a hold-down period

Inactive routes in DYMO are subject to a timeout. This timeout insures that routes that are unused for a given period of time are 'refreshed' before use. This refreshing process is identical to a route discovery attempt utilizing RREQs and RREPs. During route discovery attempts originator nodes also have a timeout to handle 'lost' RREQs (Figure 33). 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 (Figure 33). This back-off between attempts increases the likelihood of the 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 33). The duration of this holddown state is configurable. Holddown states only affects route discovery for 'failed' targets. To summarize, the following aspects of DYMO's behaviour are configurable: The time to wait for an RREP, the maximum number of discovery attempt for a target and the duration of a holddown (Figure 33).

3.2.6 Other Features

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

Router Clients

All DYMO nodes discussed thus far are routers, in that 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 blindly to a 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 reduces route message traffic and

the workload of clients. However, as with the CubeMac's master slave relationship, clients can become isolated from the network if their associated router fails.

Multicast RREPs

DYMO uses multicasting in place of AODV's broadcasting. In this work all network interfaces are configured to belong to the same multicast group. This configuration makes DYMO's multicast behaviour identical to that of broadcasting. As such, 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, this work refers to DYMO's multicasting as broadcasting.

RREQs are broadcast by default. RREP can also be optionally sent via broadcast rather than unicast. The primary benefit of this feature is that it reduces the overall number of RREQs that network nodes will generate for the given target. As with RREQs, nodes which pass on RREPs will use the RREP to opportunistically populate its routing table. As such, this feature may result in nodes adding routes they will never use. This can be especially wasteful in large networks.

Hop Limits

As DYMO RREQs traverse a network via broadcasting nodes update a 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 its 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 N then any RREQ which makes more than N hops without reaching a target can be safely dropped.

3.2.7 D3 Modifications

In the case of D3, DYMO is comparatively unaltered when compared to CubeMac and C/TDMA. D3 introduces two inter-connected additions; The concept of a ground master and the special treatment of multiple interfaces (I/Fs). In simulation all CubeSats with a CSN have two interfaces, one for S2S and the other for S2G communications. 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 the S2G I/F is used, then no S2S routes will be discovered. Also, the I/F on which an RREP arrives is used as the I/F for all traffic for the RREP's target and originator. 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, which is not desirable. These issues are overcome by introducing the role of ground master GM. 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 designate a master as GM and the impact thereof is discussed in section 4.1.1.2.
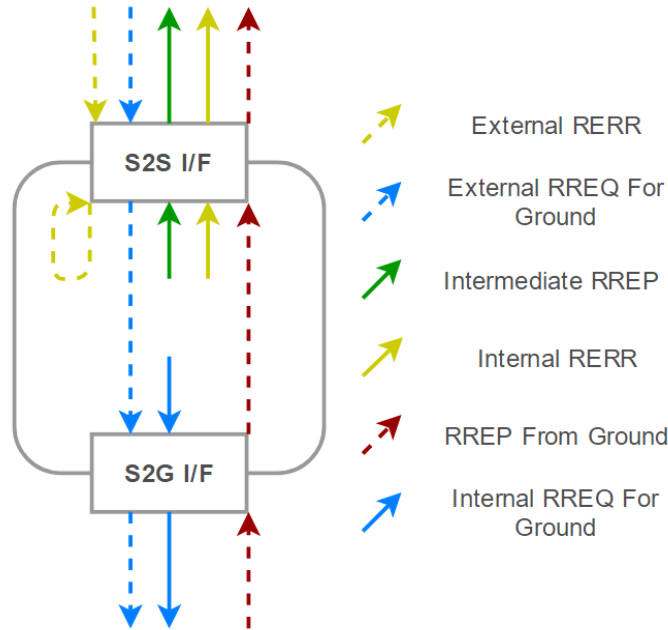
Figure 34. The interface use of a GM to all route messages implemented by this work. In comparison non-GMs will only ever use their S2S interfaces.

The GM role allows D3 to overcome the aforementioned multiple interfaces issues by changing the default behaviour of GMs. 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 34). GMs must also change their default route storage approach to insure that the next hop address in routes to ground reference the correct I/F. Non-GMs effectively ignore their S2G I/F entirely and use their S2S I/F for all communications. The final implementation of the GM role required many small changes to the existing OMNeT++ implementation of DYMO. All salient effects of these changes relate to the aforementioned I/F behaviours.

3.2.8 Drawbacks

D3 has several undesirable characteristics. Most notable among these is its inability to retain redundant routes. This was a clear design choice by the designers of DYMO as there are other AODV derivatives which allow nodes to maintain redundant routes for a given target. Protocol designers must weigh the cost of a node storing redundant routes against the overall network cost of requiring a complete route discovery attempt every time a route breaks. Within this work's hypothetical mission only routes to ground can break. As will be discussed further in relation to simulation scenarios, it is assumed that only one master may be a GM at any one time. As such, all valid routes to ground at any one time 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 benefits in maintaining 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 nodes 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 networks. Such information, when opportunistically shared amongst nodes, can greatly benefit the efficiency of route discover, maintenance and choice. Position based routing approaches have become popular in recent VANET related research [122]. As discussed in section 4.1.1.2, position information is utilized in GM designation. Aside from this use, D3'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 D3 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 a link created using an S2G I/F and S2S I/F. An appropriate cost function should 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 when no other link is available. Also, multiple nodes may establish S2G links. The cost functions for these links could reflect a node's remaining battery capacity, number of queued packets and expected S2G communication. In this case there is a clear case to maintain redundant routes to ground. The primary drawback of this approach is the added energy expense of conducting route discover over multiple S2G I/Fs when, in reality, only a small number of these will succeed. Ultimately, without an implementation of this approach to compare to the GM approach it cannot be stated empirically which is likely to be more performant.

Chapter 4: Simulation

OMNeT++ is a discrete event network simulator. An OMNeT++ simulation is constructed from a number of interconnected modules. Modules are written primarily in C++. OMNeT++'s own "ned" syntax is used to further describe C++ modules and define their interfaces and parameters. Modules may represent any logical or physical element within a network such as a router, software application, network connection or network protocol. Messages are passed between these modules in response to scheduled events. Messages are described through the use of OMNeT++ "msg" C++ template files. Through inter-module message passing OMNeT++ can be used to simulate 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 35). A number of scenarios which modify this base simulation are presented. Modules which perform the actions of CubeMac and D3 are included in the base simulation. As far as possible, the salient operation of CubeMac and D3 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.
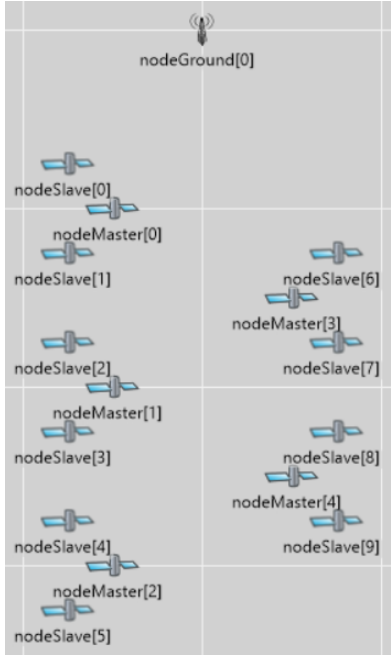
Figure 35. A visual rendering generated by OMNeT++ of the network implemented in this work's base simulation. As illustrated in Figure 4, the simulated motion of "nodeSlaves" and "nodeMasters" will bring the nodes over ground. Video representing the network's behaviour has been made available online [].

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 4 and Figure 35. Other notable scenario commonalities are as follows:

- All masters and slaves generate numerous 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.

  o 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 approximately 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 35).

- 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 implementation of CubeMac and D3. 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 [].

4.1 Implementation

Figure 36. INET's "DYMORouter" module's various components and parameters. The module shown represents nodeMaster[0] (Figure 35). Several parameters are omitted.

This works makes extensive use of the INET framework for OMNeT++ [123]. The INET framework is directly integrated within OMNeT++ releases. All relevant modules within this work's simulation scenarios are available through INET, with the exception of the D3 and CubeMac's modules. For instance, an INET module "DYMORouter" represents a network node such as nodeMaster[0] or nodeGround[0] (Figure 35). This module in turn contains several INET modules which further describe the node's behaviour and properties. As illustrated in Figure 36, a module named "dymo" is contained within the DYMORouter module named nodeMaster[0]. This DYMO module has been adapted from an existing INET module in order to perform the operations of D3. Two interfaces modules are also present; "wlan[0]" and "wlan[1]". In this case 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 37).

Figure 37. The parameters of and modules within nodeMaster[0]'s "wlan[0]" module. Wlan[0] constitutes nodeMaster[0]'s S2S interface.

In response to a scheduled event, nodeMaster[0]'s "UDPBasicApp" module will generate a 128 Byte UDP packet addressed to nodeGround[0]. This packet is passed to the "IPv4NetworkLayer" module. In the case that no route to ground can be found within the "IPv4RoutingTable" module the packet will be passed to the "DYMO" module, which implements D3. D3 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 D3's internal queue and sent to the appropriate interface (Figure 34). 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 send 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 an S2S interface of another network node. The packet will be passed from the reciever 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 38.
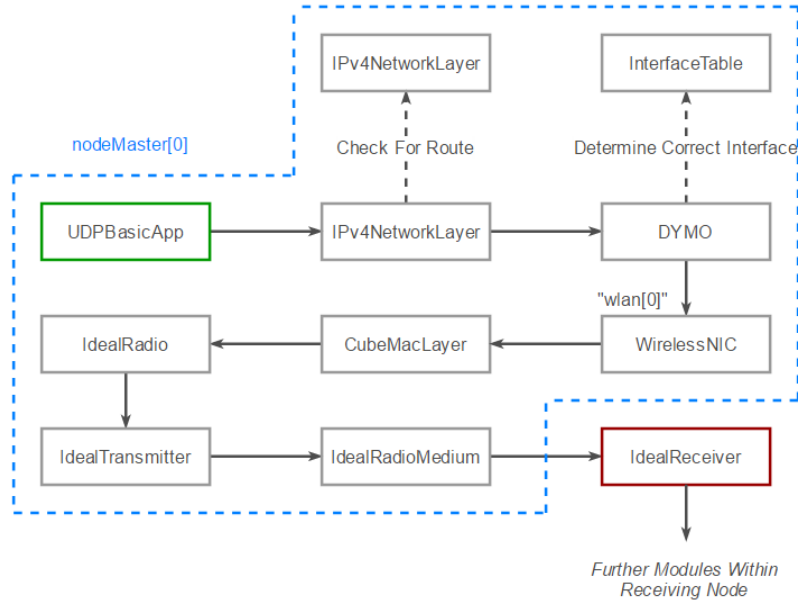
Figure 38. An example packet's progression through various modules within nodeMaster[0] which culminates in its reception at another node within the network. This path represents a common packet progression which includes both D3 (The "DYMO" module) and CubeMac.

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 D3 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 the LEO environment. 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 module modifications required by CubeMac and D3. Such issues are discussed further in section 4.2.

4.1.1 CubeMac

Figure 39. The various visible elements of a CubeMac module. From top to bottom are: Parameter, gates (connection points between modules), a vector (for result recording), watched internal module variables and owned messages. The above messages are all used for internal timing.

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 and all other salient CubeMac aspects remain. The behaviour of D3 is unchanged and only node originally designated as masters may conduction S2G communication . CubeMac is compared to its pure TDMA module through scenario 2a.

C/TDMA's time slots are modelled by CubeMac modules 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 comparatively trivial. All slaves enter a "send data" state during the uplink slot and send packets to the IdealRadio module (Figure 38). 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 D3
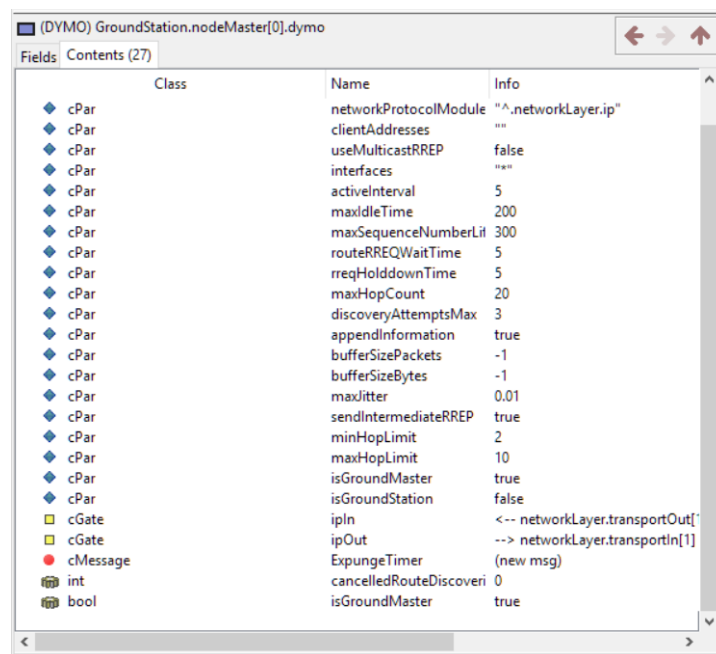
Figure 40. Several parameters such as "sendIntermediateRREP" relate to D3's modifications of DYMO as described in section 3.2.

D3 retains the majority of the core INET implementation of DYMO. Any references to a DYMO module in simulation scenarios should be taken to refer to the D3 protocol.D3 implements two new parameters: "isGroundMaster" and "isGroundStation". These parameters are used to handle the modifications referred to in section 3.2.7. D3 is implemented in a more functional manner than CubeMac. D3 specifies dedicated functions for handling each type of route message. D3'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 module. Most notably, RERRs were non-functional, link costs and sequence numbers were incorrectly implemented and no mechanisms for loop detection were implemented. Of these, link costs, sequence numbers and loop freedom were made operative in D3. 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 D3 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 this work's simulation of proposed protocol. 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 simulation.

D3'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 network nodes and determines which node is currently the GM. If the GM is not the node closest to ground the oracle will call custom functions to transfer the GM role to the closest master (Figure 41). In simulation scenarios, at least one master is always in 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 generate a RERR. However, as discussed, the implementation of RERRs in D3 is fundamentally broken. In place of RERRs propagating throughout the network, the oracle calls a function for each slave and master. This function instructs the node to drop all routes to ground and cancel any ongoing route discovery attempts.
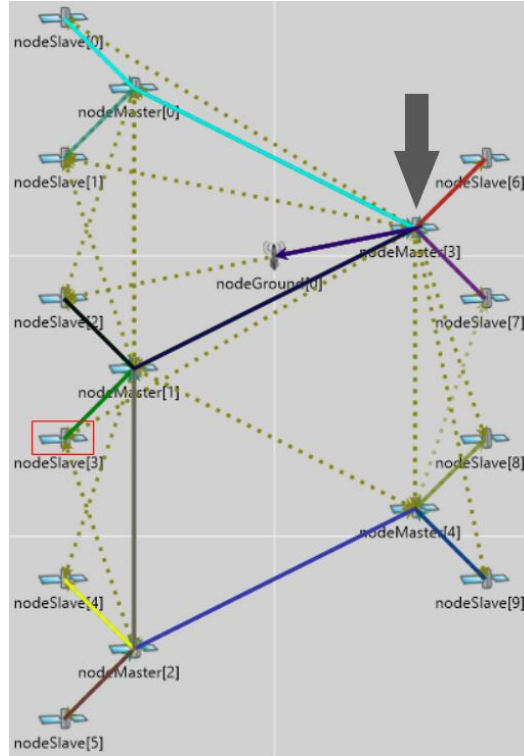
Figure 41. The closest master to ground, indicated by the grey arrow, is elected as the network's GM. The paths of UDP packets are indicated by coloured solid lines. Logical routes determined opportunistically as a result of the movement of route messages are indicated by dotted line.

Figure 41 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 a number of possible causes of this route choice. Rather than a hop count based link cost function (Figure 32) D3 implements a function based on a receiving node's energy consumption since the beginning of simulation. This link cost allows D3 to favour routes with nodes which have consumed the least amount of energy. However, this cannot be the cause of the route choice in Figure 41 as the available route to nodeMaster[3] is necessarily cheaper. All link costs must 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 D3. In this case

nodeMaster[4] began route discovery and received an intermediate RREP from nodeMaster[2]. A RREP from ground should also have reached nodeMaster[4] and allowed it replace its more expensive existing route to ground. It is possible that D3's use of sequences numbers caused nodeMaster[4] to drop the cheaper 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 37, control various aspects of operation of differing modules. The parametrization of the 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 affects other than detecting whether a two radios are in 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 39
DYMO	*	*	Figure 40
IdealTransmitter (S2S)	communicationRange	125m	Unrealistic. Adjusted to improve visualization
IdealTransmitter (S2G)	communicationRange	150m	Scaled from 750km

LinearMobility	intialZ	100m	Scaled down 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 simulation design and configuration as discussed in the preceding sections. Scenario 1b is a slight modification of Scenario 1a wherein certain additional CubeMac energy saving features, described in section 3.1.5, are disabled. Specifically, "No Data" and "Last" packets are disabled. The only configuration change made by Scenario 1b is to set 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 pureTDMA parameter set to "true", Figure 39. All slaves have their parameters adjusted to match those of master nodes. Scenario 2b, replaces the CubeMac module in S2S I/F modules 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 simulation start. The "distance-energy" approach calculates a score for each master. A lower score indicates a more favourable master. If a master with a lower score is within communication range of ground it assumes the GM role. Each score is calculated as:

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

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

Energy ranks are calculated as one plus 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 weight of 0.3 was selected.

4.2 Issues

The design and implementation of simulations 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) [124]. However, it became apparent that many key 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 seen as a more suitable tool for the development and assessment of simulation scenarios. Nonetheless, several significant issues arose development using OMNeT++ and its INET framework.

4.2.1 CDMA

Simulations are configured to use INET's IdealRadio modules (Figure 38). These modules inherit the majority of their code from the INET "Radio" module. The Radio module is extended and accessed by numerous common 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 the reception packet. 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 prior to the 'end' of an ongoing receptionTimer, only one reception will succeed due to the 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 CDMA issue broke the majority of radio modules based on the Radio module other than the IdealRadio module. This forced a development choice to either continue attempting to modify the INET modules representing more realistic radios and radio media 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 relating to estimates of the time taken for route message to traverse a given network. Due to the asymmetrical TDMA based nature of CubeMac even highly accurate estimates resulted in numerous instances of packet loss, repeated route discovery failures and runtime errors. Although it was found that, through multiple parameter sweeps, 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. It is suspected that many of the routing module related issues were partially a result of differing development rates across the open source OMNeT++ and INET framework projects.

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 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 3.27 omits several challenges which had to be addressed in order to implement the desired multiple interface behaviour (Figure 34). By default, DYMO is implemented to announce, a single "routerID". In a majority of cases, this ID is added to the address field of 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++'s initialization phases. However, the use of the addresses of both interfaces are required. Several workarounds were necessary in order 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 is poorly suited for application with even simple multiple interfaces 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 a number of modules to correctly handle 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. Despite this, it was possible to avoid the majority of these issues. However, one issue arose near the end of the development of the final 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 from this work's simulation scenarios. Each of the five scenarios is run for 810 seconds in simulation time. This 810 second period represents three consecutive passes over ground as described in opening section of chapter 4. OMNeT++ is configured to record a numerous metrics during each scenario run. These metrics can be used to compare the performance of 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 packet sizes are fixed at 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 be 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 packet per second saturation rate to accommodate the effects of ground master election and on-demand route discovery.

D3's route messages are excluded from received packet counts. However, the effects of D3's activities on not energy consumption are not omitted. The only simulated sources of energy consumption are node radios The notable parameters for radio modules and submodules are presented in Table 2. Node's do not recharge their energy stores as they

would in a realistic mission. As all scenarios utilize the same base orbital formations and velocities 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. D3 and CubeMac are utilized in their default states as described in chapter 3. It is important to note that D3's oracle is using the aforementioned "closest-master" election approach and CubeMac has all additional energy saving features enabled. The results presented in this section are primarily intended for comparisons results from this work's other scenarios.
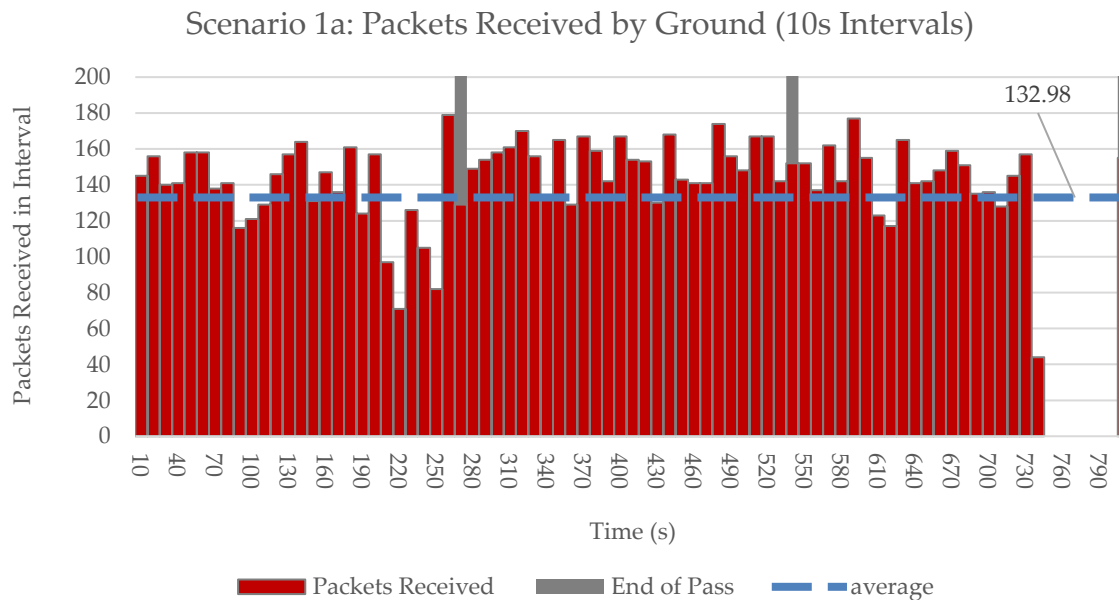
Figure 42. The number of packets received by ground station over time reduced to a granularity of ten second intervals. Each pass lasts 270s. The end of each pass is represented by a grey column.

107

Two anomalies are illustrated in the results shown in Figure 42 both of which relate to periods when 'nodeMaster[2]' occupies the GM role. NodeMaster[2] is the last master in the CSN to obtain the GM role when D3 is configured to elected the closest master as GM (Figure 41). Using this election approach, nodeMaster[2] will hold the role for ~85 seconds beginning at the 185 second mark of each pass (*Table*). 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 42). While nodeMaster[2] is the GM (725s – 810s) during the third and final pass, packet reception drops to zero for 60 seconds (Figure 42). These drops are not expected features of D3 and CubeMac's combined behaviour. Neither protocol module reported unexpected states or erroneous behaviour and the second pass shows no signed 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 these anomalies. This possibility is strengthen by the fact that this anomaly is not present in 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.

Table 3 Each master's start time as ground master during a pass and duration spent as ground master for a single pass. These times reflect the closest-master default election approach of D3's oracle and the orbital formation of nodes (Figure 41).

Master	GM Start Time	GM Duration
0	0s	85s
3	85s	35s
1	120s	30s
4	150s	35s
2	185s	85s
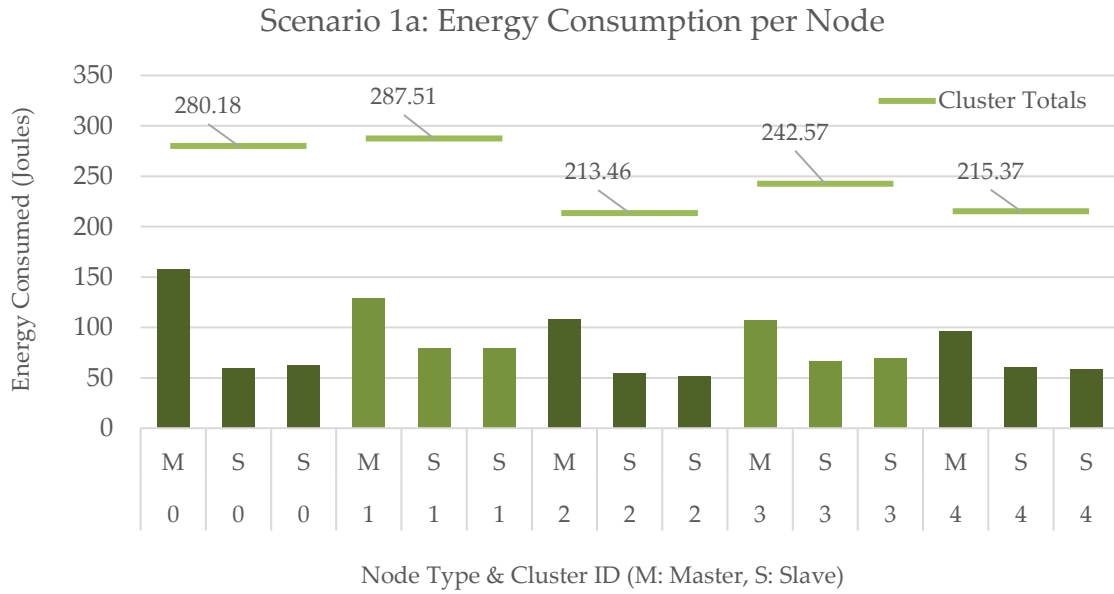
Figure 43. The total energy consumed per node over scenario 1a's simulation run. As masters must handle all slave packets and intermittent S2G communications their energy consumption is notably higher. Masters consumed an average of ~120J and slaves consumed an average of ~64J.

As expected from the design of C/TDMA and CubeMac's additional energy saving features, master nodes consume more energy overall than slave nodes. It is also evident from Figure 43 that the master of cluster 0 (nodeMaster[0]) consumes the most energy overall. This is expected as this master hold the ground master role for the longest possible duration through closest-master election (Table 3). Without the aforementioned 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-randomly sampling a of exponential distribution as described in section 4.1. This results in an uneven distribution of the total number of packets generated per node. This is the likely cause of fluctuations slave energy consumption results. Within a single cluster slave energy consumption results are closely matched. As there are only two slaves per cluster this cannot be asserted

as correlation. However, 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 41). This centrality results in the cluster handling a greater number of D3 route messages which increases master-slave 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 overall (90s). Despite this, nodeMaster[1] has the second highest level of energy consumption. This is a direct result of cluster 1's centrality. NodeMaster[1] is more likely to be included in active routes to ground which increase the node's total energy consumption.

5.2 Scenario 1b

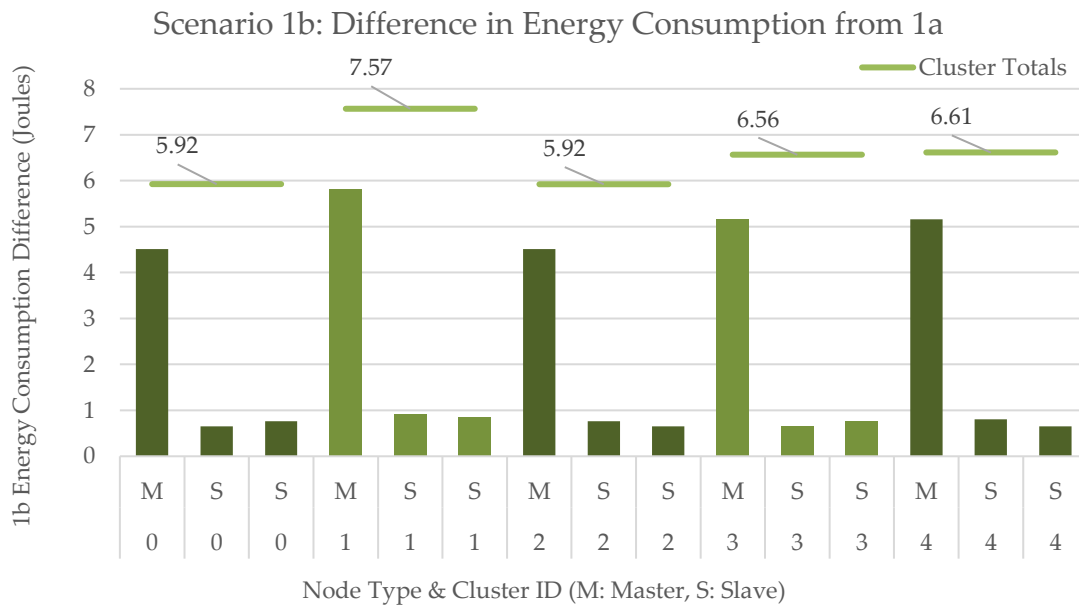Scenario 1b: Difference in Energy Consumption from 1a

Figure 44. Energy differences calculated as the energy consumed by each node in scenario 1a less the corresponding node's energy consumption 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, including the aforementioned nodeMaster[2] anomaly. Scenario 1b's packet results are identical to those of scenario 1a. As such, this section omits these results. 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 43 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. Changes in energy consumption show a low variations 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 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%. These results illustrate that CubeMac's energy saving features success in reducing energy consumption without impacting S2G throughput.

5.3 Scenario 2a

CubeMac's pure TDMA mode removes the concept of node clusters by assigning all nodes to the master role. This change requires that the slave uplink be dropped from each CubeMac frame. The total number of slots must then extend from 5 to 15. Scenario 2a is parameterized to insure that nodes which were previously slaves exhibit the behaviour of masters. All other aspects of CubeMac and scenario 1a are unchanged. D3's behaviour is also unchanged and it will only elect GMs from the original group of masters.
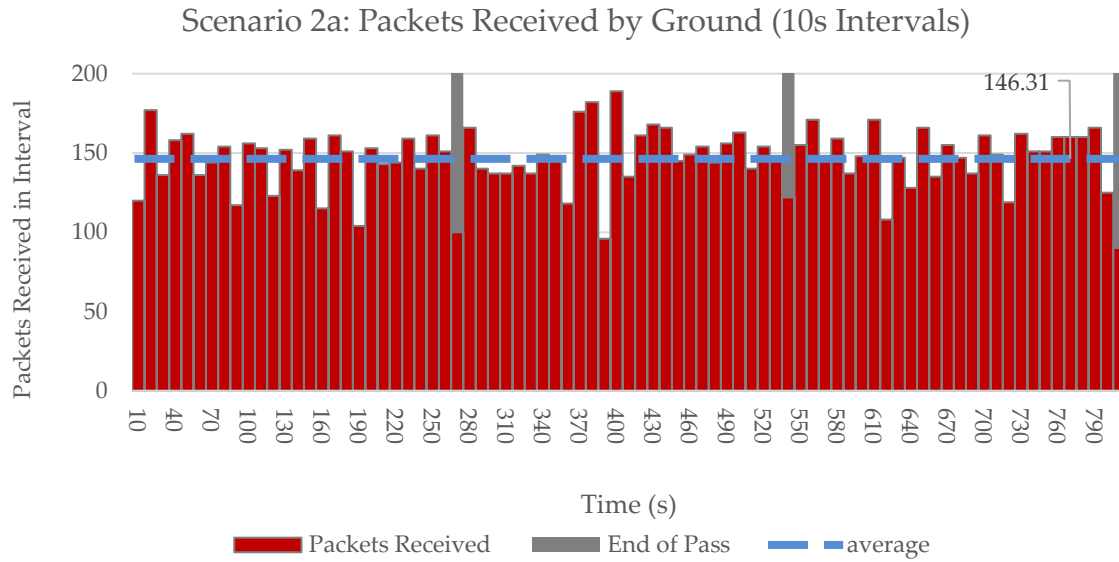
Figure 45. CubeMac's pure TDMA ground station packet reception performance. When compared with Figure 42 CubeMac's pure TDMA mode can be seen to provide more consistent throughput.
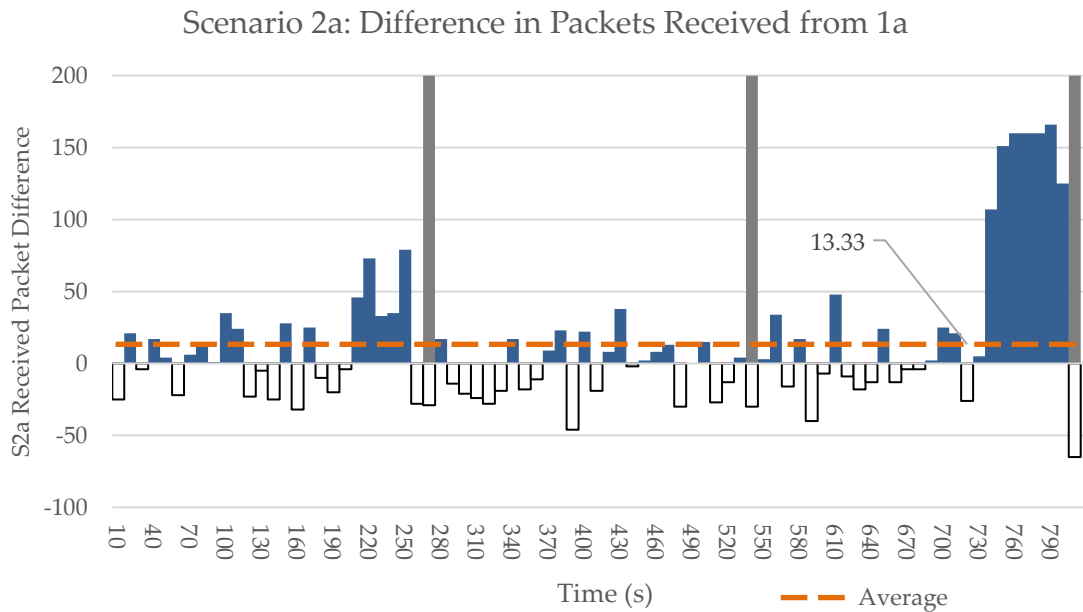
Figure 46. The difference over time in the number of packets received by ground in scenario 2a as compared with scenario 1a. Negative values (outlined) represent interval values in which scenario 1a's ground station received more packets than scenario 2a's ground station.

It is evident from Figure 45 and Figure 46 that pure TDMA does not suffer from the aforementioned scenario 1a anomaly which suggests the anomaly is related to the behaviour of CubeMac slaves. Scenario 2a has a total packet reception of ground result of 11851 which is an increase in 1080 packets over scenario 1a. After adjusting for the scenario 1a anomaly this increase falls to ~120 packets. This represents a ~1% increase over scenario 1a, however the statistical significance of this adjusted result is dubious. To examine the true S2G throughput performance difference between scenario 1a and 2a requires the resolution of scenario 1a's anomaly.
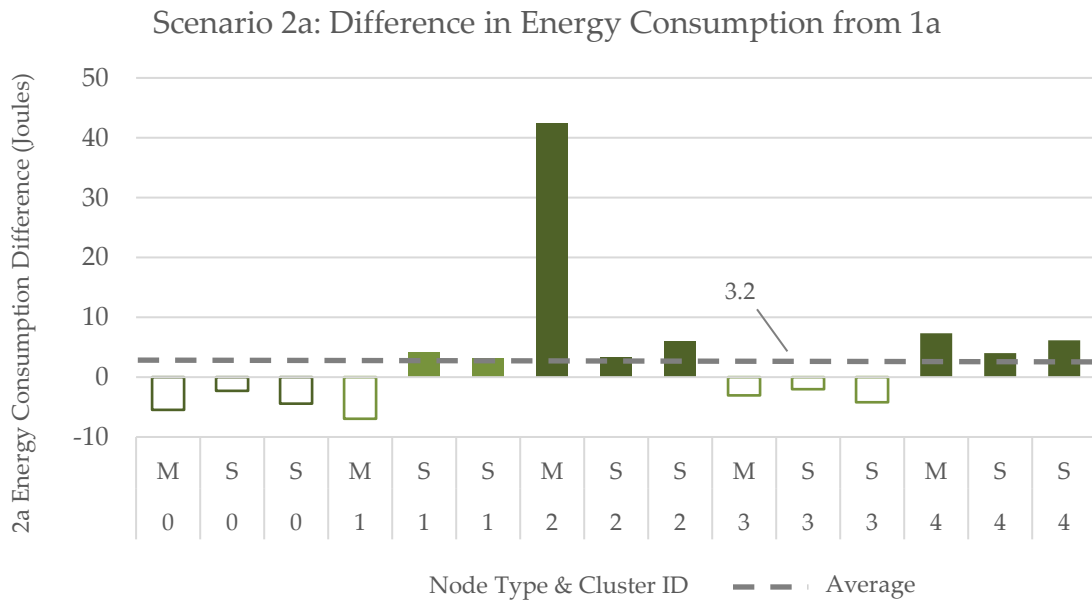
Figure 47. The change in energy consumption of individual nodes in scenario 2a as compared to scenario 1a. Negative (outlined) columns represent nodes which, in scenario 2a, consumed less energy overall than corresponding nodes in scenario 1a.

As with packet reception results, difference in scenario 2a's energy consumption results are effectively nullified by scenario 1a's anomaly. As expected, Figure 47 shows that nodeMaster[2]'s energy consumption increases significantly when it communicates with ground for the expected 255s duration. All other changes in node energy consumption are

likely 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 48).
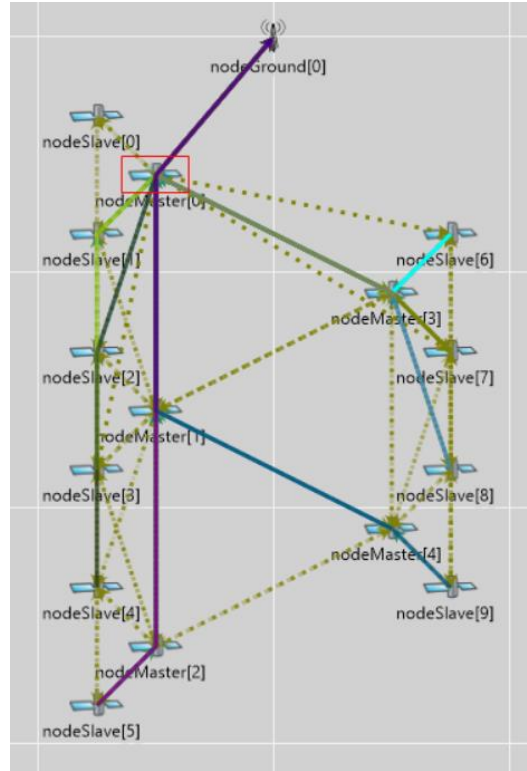
Figure 48. In CubeMac's pure TDMA mode all nodes act as masters. However, only 'nodeMaster's may form S2G links. This arrangement creates a larger number of viable routes to ground. Solid lines above represent the movement of science data packets. This behaviour can be compared to the default CubeMac & D3 behaviour as illustrated in Figure 41.

5.4 Scenario 2b

Scenario 2b replaces the custom CubeMac module entirely with an existing INET module which implements a CSMA based protocol. As with scenario 2b as few elements of scenario 1a are altered as possible. Where possible, the parameters of the utilized CSMA module are matched to those of CubeMac in scenario 1a.
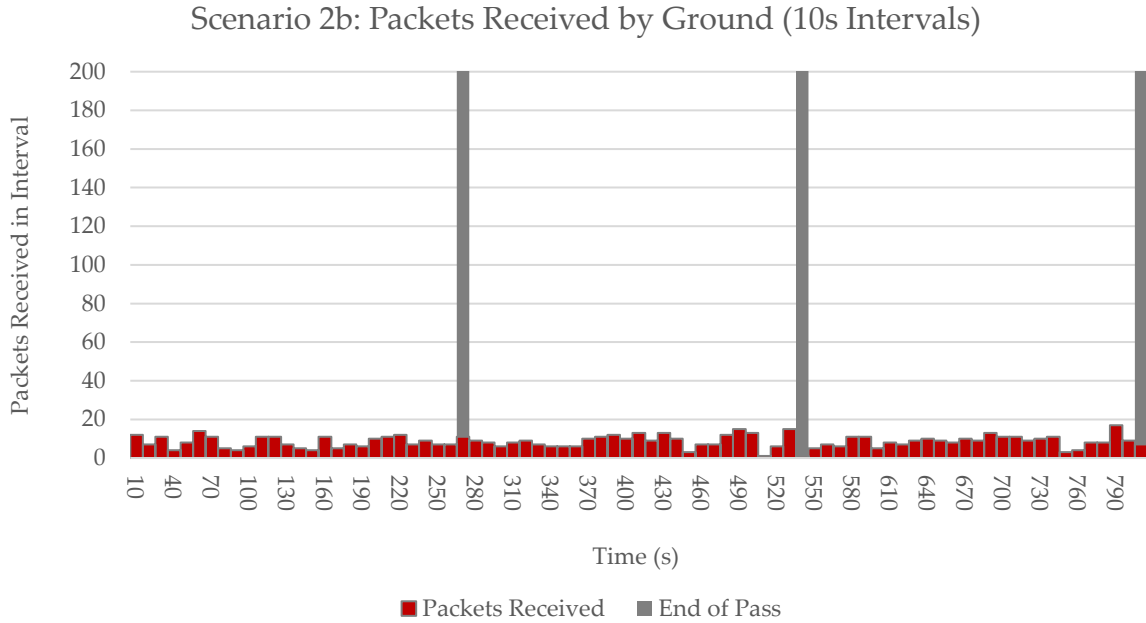
Figure 49. Introducing the existing CSMA MAC protocol results in an order of magnitude drop in the number of packets received by ground. This figure can be compared to both Figure 42 and Figure 45 which use the same y-axis bounds.

Figure 49 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 in comparison to scenario 1a (Table 4). Radhakrishnan et al. report a total 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%. Where throughput here is measured as total number of packets received at ground as a percentage of the total number of generated packets. 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 such as CSMA. However,

the considerable gap between scenario 1a's and scenarios 2b's performance calls into question the fidelity of scenario 2b's results.

TIDY UP Considering the number and frequency of INET related issues encountered by this work it is entirely feasible that the utilized CSMA module has fundamental flaws. Without 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 D3. Route discovery in D3 relies on flooding the CSN with RREQs through a series of rebroadcasts. In this work, all nodes will attempt route discovery at approximately the same time. This is caused by the D3 oracle's instantaneous removal of all routes to ground in response to a change in ground master. Simultaneous route discovery attempts by all nodes will result in numerous collisions. As such, route discovery will be slowed dramatically. Slowing route discovery increases the chances of node's timing out their RREQ's and observing a back off period. Overall, the use of CSMA greatly reduces the ability of nodes to obtain valid routes to ground. Increasing the time required to obtain a route to ground reduces the effective window a node has to communicate its packets to the current GM which reduces possible data packet reception at ground.

5.5 Scenario 3

…
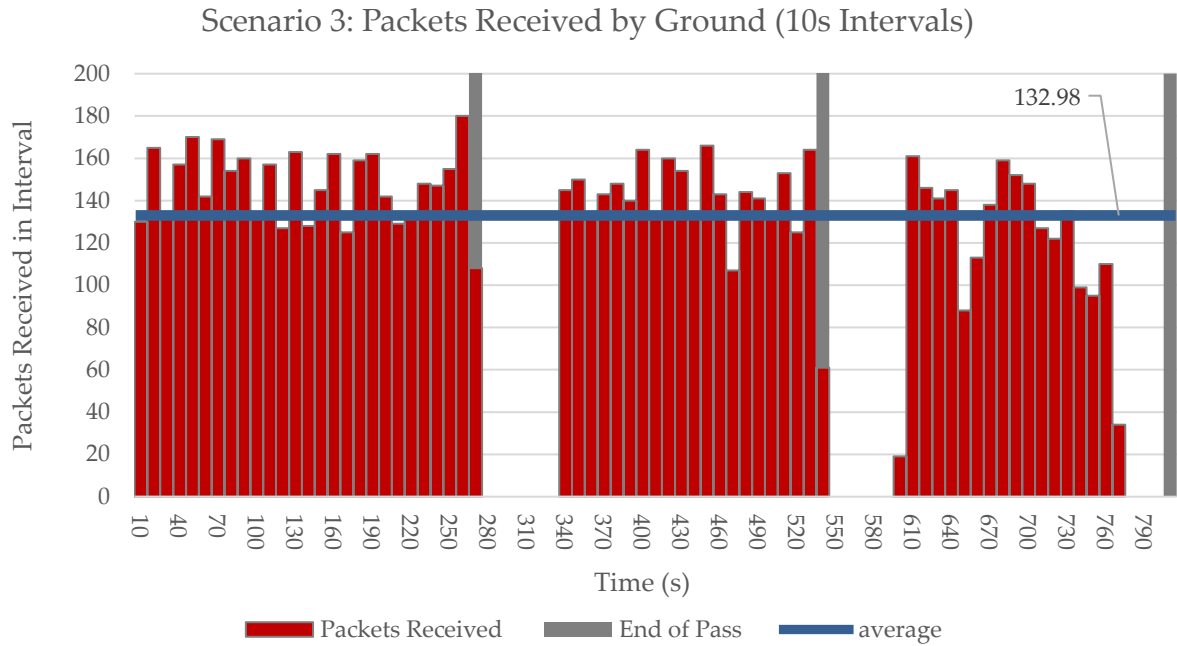
Figure 50. Scenario 3 imposes greater restrictions on access to ground through an adjusted ground master election approach as discussed in section 4.1.6. Scenario 3 does not suffer from the anomalous behaviour of scenario 1a.

…

Scenario 3: Difference in Packets Received from 1a
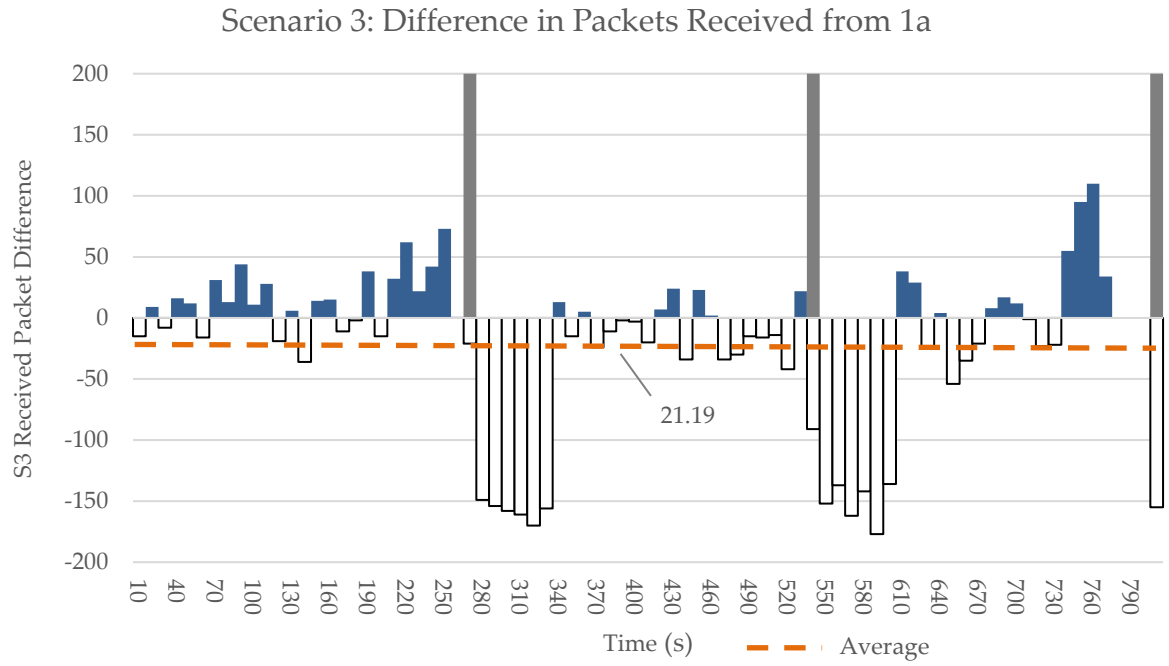
Figure 51. The difference in received packets in scenario 3 as compared with scenario 1a. Restricted ground access results in clear reductions in throughput during the opening periods of the second and third passes.

…
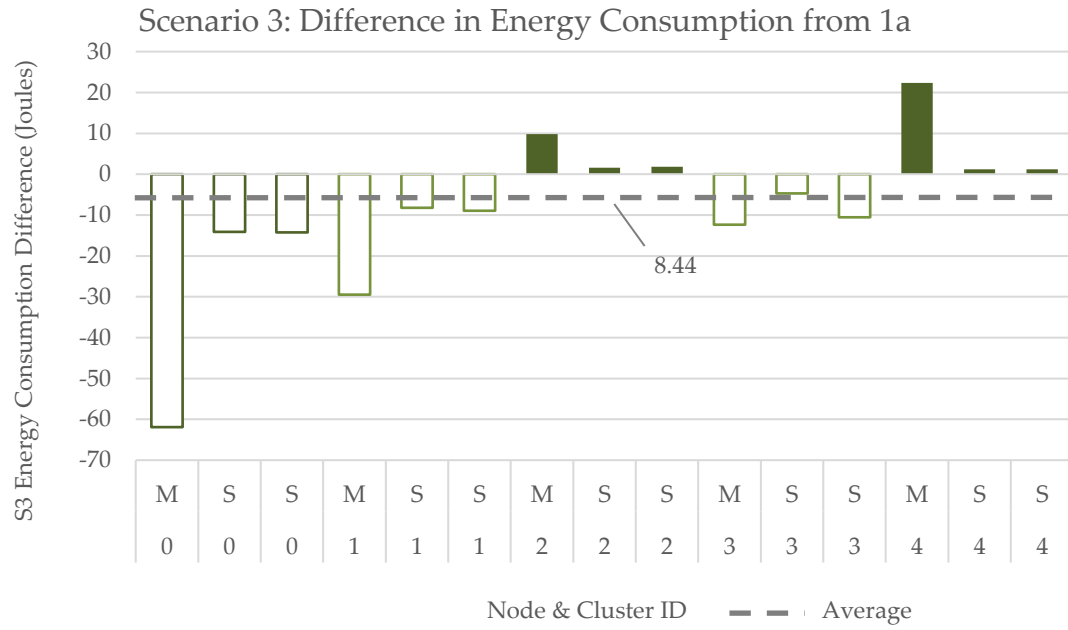
Figure 52. Negative (outlined) columns represent the energy saved by individual nodes as a result of altered D3 ground master election.

…
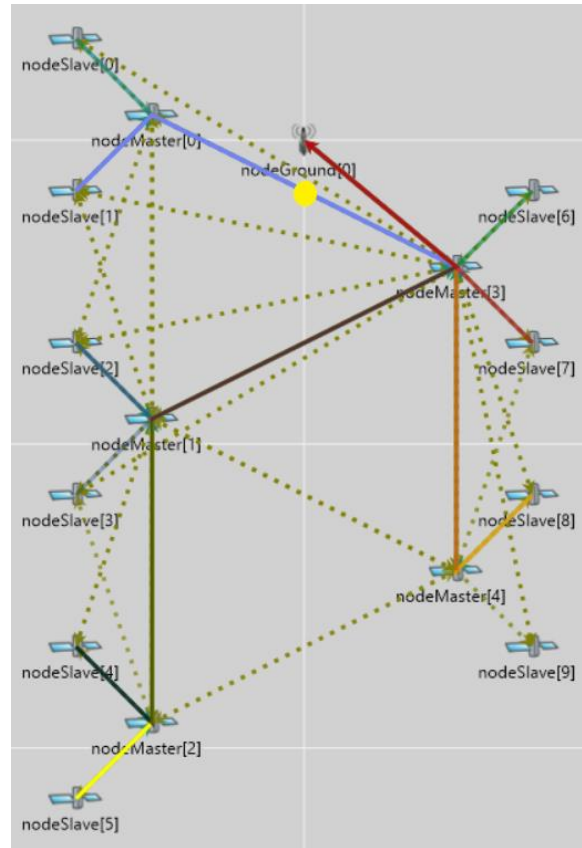
Figure 53. ...

…

5.6 Efficiency

… CDMA additional cost on the conservative side …

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

… Summary of key points

6.1 Discussion

…

6.1.1 Results

… Avoid summary

6.1.2 Contributions

…

CubeMac

…

D3

…

Simulation

…

6.1.3 Space Junk

…

6.1.4 Mission Design

Those familiar with mission planning and satellite technology will note that there is little treatment of mission operational requirements and technology readiness. These omissions allow a focus on the PvTP trade-off by ignoring many of the realities mission design. In reality, space bound technology must be thoroughly understood, tested and generally as simplified as possible. 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 clear benefits; It allows for complete knowledge of the protocol by mission designers and operators and it greatly reduces the risk of the existence of unknown bugs or flaws.

Once a satellite is in orbit there is little that can be done to fix issues. As such, mission designers tend to focus on simple, easily-understood and if possible space-tested approaches. Both CubeMac and DYMO are less complex than some current MAC and routing protocols but, there are definitely less complex approaches in existence. For instance, TDMA and DSR. These approaches may not be as performant but it would be simple to test in simulation prior to launch and issues in orbit would be easily understood.

This preference towards simplicity provides part of the motivation behind protocol design choices, especially with CubeMac. There are several protocols such as LDMA which could have provided a basis for CubeMac [9]. In many cases it was felt that the added complexity of these protocols was not accompanied by sufficient benefits.

6.2 Future Work

…

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

[4]   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.

[5]   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.

[6]   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.

[7]   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.

[8]   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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

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

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

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

[17] 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.

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

[19]   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.

[20]   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.

[21]   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.

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

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

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

[25]   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.

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

[27]   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.

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

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

[30]   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.

[31]   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.

[32] J. Farkas, "CPX: Design of a standard cubesat software bus," *California State University, California, USA,* 2005.

[33] 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.

[34] 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.

[35] K. Hayward, "The Economics of Launch Vehicles: Towards a New Business Model," in *Yearbook on Space Policy 2015*, ed: Springer, 2017, pp. 247-256.

[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] 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.

[124] G. F. Riley and T. R. Henderson, "The ns-3 network simulator," *Modeling and tools for network simulation*, pp. 15-34, 2010.