

DELAY TOLERANT NETWORKING FOR CUBESAT TOPOLOGIES AND
PLATFORMS

By

PAUL MURI

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2013

© 2013 Paul Muri

To my beloved parents, Theresa Muri and David Muri; and to my colleagues, friends,
and family who believe in me

ACKNOWLEDGMENTS

I would like to express my sincerest gratitude to my academic advisor, Prof. Janise McNair, for her invaluable guidance and continuous support throughout my undergrad and graduate studies at the University of Florida. Her openness and encouragement of my ideas helped me become a better researcher. Without her excellent advice, patience, and support, this doctoral dissertation would not be possible.

I would like to thank all of my committee members, Prof. Jenshan Lin, Prof. Jasmeet Judge, and Prof. Norman Fitz-Coy, for serving on my committee. The committee's valuable comments and constructive criticism helped to improve this dissertation greatly.

I also would like extend my appreciation to the Wireless and Mobile System laboratory colleagues. They always gave me valuable suggestions, generous support, and fruitful "jam sessions" during my graduate studies at the University of Florida. I thank all of my labmates, Gokul Bhat, Obul Challa, Max Xiang Mao, Sherry Xiaoyuan Li, Jing Qin, Jose Alomodovar, Krishna Chaitanya, Dante Buckley, Ritwik Dubey, Bom Leenhapat Navaravong, Dexiang Wang, Jing Jing Pan, Gustavo Vejarano, Seshupriya Alluru, Eric Graves, and Corey Baker, in no particular order. All of them made my days in the lab more enjoyable.

This work was supported by a NASA Office of the Chief Technologist's Space Technology Research Fellowship (NSTRF) grant number NNX11AM73H. I would like to thank Dave Israel, Tom Flatley, Gary Crum, Jane Marquart, Greg Menke, and Faith Davis at NASA Goddard Space Flight Center for providing equipment and expertise.

The work from Chapter 3, and Chapter 4 was partially supported by IRAD grants from Lockheed Martin's Information Systems & Global Services (LM IS&GS) group, and the Advance Space Technologies Research & Engineering Center (ASTREC).

This work was also partially supported in part by the National Science Foundation (ECCS-0901706). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Last, I would like to thank my parents, Theresa Simpson Muri and David Muri, for their love, support, and sacrifice. They have always been my role models of hard work and a constant source of inspiration and encouragement throughout my life.

TABLE OF CONTENTS

	<u>page</u>
ACKNOWLEDGMENTS	4
LIST OF TABLES	9
LIST OF FIGURES	10
LIST OF ABBREVIATIONS	13
ABSTRACT	17
CHAPTER	
1 INTRODUCTION.....	19
Distributed Satellite Systems and CubeSats	19
List of Major Contributions	20
Radio Frequency Allocation	22
Application and Orbital Properties.....	23
Delay Tolerant Networking	24
Optical Communications	25
Organization.....	25
Communication Sub-systems for Intersatellite Systems and CubeSats.....	26
Enhancing CubeSat Communication Through Antenna Design.....	26
Topology Design and Performance Analysis for Networked LEO Satellites	27
Simulating Delay Tolerant Networking for CubeSats	28
Performance of DTN Protocols for Delay Space-based Optical Channels	29
2 A TAXONOMY OF CUBESAT TRANSMITTERS, LINKS, AND LAUNCHES	30
History of Intersatellite Linking Communication Sub-systems	30
1972, 74, 78 OSCARs 6,7,8.....	31
1976, LES-8 and LES-9.....	32
1983, TDRSS	33
1997, Iridium.....	34
History of CubeSat Communication Sub-systems.....	35
2003, Eurockot LV, Pletsak, Russia	36
2005, SSeti Express, Pletsak, Russia	39
2006, M-V-8, Uchinoura, Japan.....	41
2006, Minotaur-1, NASA Wallops, VA, USA	41
2007, DNEPR-2, Baikonur, Kazakhstan	42
2008, PSLV-C9, Satish Dhawan Space, India	43
2009, Minotaur-1, Wallops, MD	44
2009, STS-127 Space Shuttle Endeavor	44
2009, ISI Launch 01, Satish Dawan Space Center, India	46

	2010, Japanese H-IIA F17, Tanegashima Space Center	47
	2010, PSLV-C15, Satish Dhawan Space Centre	48
	2010, STP-S26, Kodiak, Alaska	48
	2010, Falcon 9-002, Cape Canaveral, FL	49
	2011, Jugnu, Satish Dawan Space Center, India	50
	2011, Elana, Vandenberg AFB, CA	50
3	ENHANCING CUBESAT COMMUNICATION THROUGH EFFECTIVE ANTENNA SYSTEM DESIGN	53
	CubeSat Antenna Trade Study	54
	Multiple Antenna System Design.....	55
	Antenna Type Selection.....	56
	Theoretical Results for Commercial Direction Antennas	56
	Deployable Helical Antenna Design.....	60
	Prototype Dimensions	60
	Impedance Matching	61
	Hemispherical Tapering	61
	Performance	62
	Power Savings	63
	Antenna Testing	64
	Commercial Patch Antenna	64
	Commercial Helical Antenna.....	68
	Prototype Helical Antenna	70
	Experiment	73
	Results	73
	Summary	74
4	TOPOLOGY DESIGN AND PERFORMANCE ANALYSIS FOR NETWORK LEO SATELLITES.....	75
	Constellation Topology Design	77
	Sun-synchronous Repeating Ground Track (SSRGT) Constellation	78
	Flower Constellation	79
	Modeling LEO Satellite Constellations	80
	Using Satellite Toolkit for Network Topology Design	80
	Communication Range Threshold.....	81
	NS-2 to Evaluate Network Performance	83
	Results	86
	Drop-Ratio Verses MAC Slot Times	87
	Throughput Versus Source Traffic Density.....	89
	Summary	91
5	SIMULATING DTN FOR CUBESATS	92
	DTN Protocol Over CubeSats	92
	DTN2.....	92

ION	93
IBR-DTN	93
JDTN	93
ByteWalla	94
N4C	94
Simulating Delay Tolerant Networking	94
Design of a Testbed.....	95
Simulation Platform	97
Virtual Machines	97
Linux Containers (LXCs).....	98
Simulated Communication Channels	98
Setup of Experiment	98
Results	100
Summary	101
 6 PERFORMANCE COMPARISON OF DTN PROTOCOLS FOR HIGH DELAY SPACE BASED OPTICAL CHANNELS	 102
DTN Implementation Performance Studies	104
Theoretical Analysis.....	107
Testbed Design	109
Architecture.....	109
Channel Emulation Software Parameters	111
Modeling Optical Atmospheric Conditions	112
Channel Emulator Experiment Setup	114
ARM Processor Test Setup.....	115
Results	116
Network Goodput for Varying Bundle Size	116
BP/TCP for High Latency	117
AT91SAM9G Resource Usage.....	118
Throughput per Bundle Size	119
Data Utilization per Transmission Unit	120
Summary	121
 7 CONCLUSIONS AND FUTURE WORK.....	 123
Future Work	124
Antennas and Radio Frequency Bands	124
Constellation Applications, Networking Parameters, and Routing Algorithms	125
DTN Standardization for Network Simulators	125
Improvements to Network Emulation, DTN Implementations, and Memory Access.....	126
LIST OF REFERENCES	127
BIOGRAPHICAL SKETCH	134

LIST OF TABLES

<u>Table</u>	<u>page</u>
2-1 History of Intersatellite Linking (ISL) Systems	30
2-2 History of CubeSat Data Transmitters by Launch Date 2003-2005.....	38
2-3 History of CubeSat Data Transmitters by Launch Date 2006-2008.....	40
2-4 History of CubeSat Data Transmitters by Launch Date 2009-2011	45
3-1 Typical values for CubeSat Transmission Parameters	54
3-2 1U CubeSat Transmitter Rates, Modulations, Frequency, and Power	55
3-3 Gain vs Dimensions for a 2.45 GHz Helical Antenna	59
3-4 Typical values for CubeSat Transmission Parameters	62
3-5 Power Savings from Helical Antenna Gain	64
3-6 Commercial Patch Antenna Specifications	64
3-7 Commercial Helical Antenna Specifications	68
3-8 Prototype Helical Antenna Specifications.....	70
4-1 Applications for Constellation Types	75
4-2 Orbital Parameters for SSRGT and Flower Constellation	80
4-3 Parameters to Calculate Transmission Range	81
5-1 Comparison of the average data rate between DTN and UDP clusters	101
6-1 DTN Space Mission Demonstrations	103
6-2 Ground Station Channel, Space Channel, and Payload Channel Parameters ..	112

LIST OF FIGURES

<u>Figure</u>	<u>page</u>
2-1 OSCARS 6 to OSCAR 7 and 8 carrier frequency bands	31
2-2 NASA's Tracking and Data Relay Satellite System	34
2-3 Iridium constellation generated by the satellite constellation visualization	35
2-4 Poly Picosatellite Orbital Deployer or P-POD holding three 1U CubeSats	36
2-5 The GeneSat-1 3U bus replicated for future CubeSats	42
2-6 Nightglow picture from Swisscube	47
3-1 Multiple Antennas on CubeSat	56
3-2 Radiation profile of a 10-element, Yagi-Uda antenna	57
3-3 Radiation profile of a 10 element Log-periodic antenna	58
3-4 Radiation profile of a 10 turn Helical antenna	59
3-5 Detailed sketch of a hemispherical helical antenna	61
3-6 The deployment for a hemispherical helical	62
3-7 Radiation profile of a 7 turn tapered Helical antenna	63
3-8 Patch antenna in an anechoic chamber	65
3-9 Patch SWR is under 2:1 from 2.3 to 2.9 GHz	66
3-10 Patch Magnitude over frequency	67
3-11 Patch Smith Chart	67
3-12 Luxul Helical SWR is under 1.5:1 from 2.1 to 3 GHz	68
3-13 Luxul Helical Magnitude over frequency	69
3-14 Luxul Helical Smith Chart	69
3-15 Prototype tapered in the anechoic chamber	71
3-16 Prototype Helical is under 2:1 from 2.3 to 2.42 GHz	71
3-17 Prototype Helical Magnitude over frequency	72

3-18	Prototype Helical Smith Chart	72
3-19	Deployable Helical Performance vs. Other Commercial Antennas	74
4-1	Series of of distributed satellite constellations and clusters	76
4-2	The SSRGT constellation.	78
4-3	The flower constellation.	79
4-4	Access time windows for sensing to sink communication	82
4-5	The network simulator wireless transmission simulation model	84
4-6	Packet drop ratio versus MAC slot time	88
4-7	Comparison of the ground to sink satellite drop ratio versus the MAC slot time	89
4-8	Throughput versus source traffic density	90
4-9	Comparison of packet drop ration versus source traffic density	90
5-1	The simulation platform screen capture of the network status (netstat)	95
5-2	A simulation platform screen capture of the python-based visualizer (pyviz).	96
5-3	Network Stack for transmitting bundles between virtual machines	97
5-4	Network Animator (NetAnim)	100
5-5	A python visualization graphical user interface of a 802.11g-2007 topology.	100
6-1	Testbed system design with convergence layers	104
6-2	Theoretical Throughput over Delay	108
6-3	Testbed System Design	109
6-4	Testbed Architecture	110
6-5	Setup for the channel emulator experiment at Goddard Space Flight Center ...	114
6-6	The AT91SAM9G board with serial cable interface	115
6-7	BP/TCPCL with varying bundle sizes vs TCP/IP	117
6-8	TCP/IP, BP/TCPCL, and BP/LTP goodput over added delay	117
6-9	Microcontroller used resources used by ION for a 65 kbyte bundle size	118

6-10	AT91SAM9G throughput for bundle sizes of 100 byte to 65 kbyte	119
6-11	Average throughput for 65Kbyte, 30Kbyte, and 10Kbyte bundles	120
6-12	Data (in Kbytes) per each transmission frame with a 9 kbyte MTU	121

LIST OF ABBREVIATIONS

AARL	American Radio Relay League
ACS	Attitude Control System
AM	Amplitude Modulation
AMS	Asynchronous Messaging Service
AMSAT	Radio Amateur Satellite Corporation's
AP	Access Point
ARC	Ames Research Center
ASIC	Application-Specific Integrated Circuit
BER	Bit Error Rate
BP	Bundle Protocol
CBR	Constant Bit Rate
CCA	Clear Channel Assessment
CCSDS	Consultative Committee for Space Data Systems
CFDP	CCSDS File Delivery Protocol
CGR	Contact Graph Routing
CSMA	Carrier Sense Multiple Access
CL	Convergence Layer
CLEO	Cisco router in Low Earth Orbit
CMR	Communication Moon Relay
COTS	Commercial of the shelf
CPU	Central Processing Unit
CSV	Comma Separated Value
DCF	Distributed Coordination Function
DGR	Datagram Retransmission

DINET	Deep Impact Network Experiment
DTN	Delay Tolerant Networking
DTNRG	Delay Tolerant Networking Research Group
FEC	Forward Error Correction
FPGA	Field-Programmable Gate Array
GEO	Geosynchronous Earth Orbit
GRGT	Guam Remote Ground Terminal
GSFC	Goddard Space Flight Center
GUI	Graphical User Interface
HPBW	Half-Power Beam Width
IMU	Inertial Measurement Unit
ION	Interplanetary Overlay Network
IP	Internet Protocol
IRIS	Internet Router In Space
ISL	Intersatellite Link
JAXA	Japan Aerospace Exploration Agency
JPL	Jet Propulsion Laboratory
LACP	Link Aggregation Control Protocol
LAG	Link Aggregation
LAN	Local Area Network
LCRD	Laser Communication Relay Demonstration
LEO	Low Earth Orbit
LLC	Logical Link Control
LLCD	Lunar Laser Communication Demonstration
LTP	Licklider Transmission Protocol

LXC	Linux Container
MAC	Media Access Control
MEO	Medium Earth Orbit
MSS	Maximum Segment Size
MTU	Maximum Transmission Unit
NASA	National Aeronautics and Space Administration
NEC	Numerical Electronic Code
NLS	Nanosatellite Launch System
NS	Network Simulator
NSP	Nanosatellite Protocol
OSCAR	Orbiting Satellite Carrying Amateur Radio
P-POD	Poly-Picosatellite Orbital Deployer
PACSAT	Packet Radio Satellite
RF	Radio Frequency
RSSI	Received Signal Strength Indication
RTG	Radioisotope Thermoelectric Generator
RTT	Round Trip Time
SAVI	Satellite Constellation Visualization
SCNT	Space Communication and Networking Testbed
SLS	Space Link Simulator
SNR	Signal to Noise Ratio
SPICE	Space Internetworking Center
SRAM	Static Random-Access Memory
SSRGT	Sun-synchronous Repeating Ground Track
STGT	Second TDRS Ground Terminal

STK	Satellite Toolkit
SWR	Standing Wave Ratio
T-POD	Tokyo Picosatellite Orbital Deployer
TCP	Transmission Control Protocol
TCL	Tool Command Language
TDRSS	Tracking and Data Relay Satellite System
UDP	User Datagram Protocol
VLAN	Virtual LAN
VPN	Virtual Private Network
WSGT	White Sands Ground Terminal
X-POD	Experimental Push Out Deployer

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

DELAY TOLERANT NETWORKING FOR CUBESAT TOPOLOGIES AND
PLATFORMS

By

Paul Muri

August 2013

Chair: Janise McNair

Major: Electrical and Computer Engineering

A CubeSat is a pico-satellite platform with a volume as small as 10cm^3 .

Universities, companies, and governments use CubeSats to conduct space technology experiments, demonstrations, and research at lower manufacturing costs. Traditionally, CubeSats communicate over amateur frequencies using point-to-point links. This severely limits both the data rate and the time window to downlink to a single ground station. However, CubeSats, as part of a spaced-based sensor network, have the potential to augment existing systems through interlinking with other spacecraft and ground stations. Through lower orbits and strategic topologies, CubeSat networks can provide data with increased temporal and spatial resolution to supplement low earth orbit missions. In this dissertation, Delay Tolerant Networking (DTN) communication protocols were designed to enable hop-by-hop store and forward communications between CubeSats. From analysis and testing, it was demonstrated that DTN protocols can be applied to address the limitations of CubeSat networks such as radio frequency link margins, topology management, intermittent connections, slow downlink data-rates, long propagation delays, and constrained processing resources. The major contributions of the dissertation are a deployable high gain S-band helical prototype,

network simulation of low earth orbits with optimized network timing parameters, a DTN network testbed with space link channel emulation, and an implementation of DTN on CubeSat hardware.

CHAPTER 1 INTRODUCTION

In 1954, the United States Navy started relaying signals off the moon. This simple reflection experiment led to the world's first space communications satellite system: Communication Moon Relay (CMR). CMR allowed Hawaii to communicate with Washington, DC [1]. For many years after, satellite systems were simple relays consisting of a ground station to satellite uplink channel, and a satellite to ground station downlink channel. Following simple relay systems, a constellation of satellites that communicated directly with one another was designed in 1972. Instead of simple uplinks and downlinks between a satellite and its ground station, these systems would use satellite-to-satellite relays known as crosslinks. Crosslinks allowed for fewer ground stations. In addition, frequencies that are quickly attenuated in the atmosphere could be used, making the link undetectable from the ground. From this technology grew the concept of distributed satellite systems (a group, or cluster of many satellites) for remote sensing and communications. However, few distributed satellite projects have been launched because of the large budget required to design, manufacture, and launch.

Distributed Satellite Systems and CubeSats

CubeSat technology is an effort to reduce the cost of distributed satellite systems. A CubeSat is a type of miniaturized satellite primarily used for university space research that has a volume of exactly one liter, weighs no more than one kilogram and is built using commercial off-the-shelf (COTS) components [2]. Since CubeSats are 10cmx10cm, regardless of length, they can be launched and deployed using a common standardized deployment mechanism. While manufacturing cost of a typical satellite weighing about 1000 kgs is an order of tens of millions of dollars, partly because of

custom made components, extensive testing, and major launch efforts. However, CubeSats were designed to provide low mass, small, COTS components, and only the payload is custom designed. As a result, a CubeSat mission tends to cost up to two million dollars, which is orders of magnitude less than the cost of a typical satellite.

For communications, CubeSats have traditionally used amateur frequencies and point-to-point links, which severely limit the data rate, as well as the time window to downlink. In this dissertation, CubeSats are considered as a network, specifically as part of a spaced based sensor network. CubeSat networks deployed to cooperate with large satellites and ground stations, allow for power efficient data flow and have the potential to provide increased temporal and spatial resolution as well as communications from very low earth orbit missions to interplanetary missions. This dissertation describes how Delay Tolerant Networking (DTN) protocols were designed and tested to achieve space-hopping communication between CubeSats. It is important to note that the applications for DTN protocols should not require real-time data, since DTN nodes queue data for periods of hours or days. To design the communication networking protocols for these scenarios, several limitations were addressed, including radio frequency link margins, intermittent connections, and slow downlink data-rates over long propagation delays.

List of Major Contributions

The major contributions of the dissertation are a deployable high gain S-band helical prototype, network simulation of low earth orbits with optimized network timing parameters, a DTN network testbed with space link channel emulation, and an implementation of DTN on CubeSat hardware. First, improvements to the CubeSat radio link margins were considered. Most CubeSats currently transmit with monopole or

dipole antennas at a low UHF bandwidth. This investigation explored several directional antenna types for allowing an S-band transmission. Novel deployable antenna mechanisms were created to address the current limitations, and analyzed for a CubeSat frame. A deployable hemispherical helical antenna was developed. The hemispherical helical antenna employed a tapered design, which provided a half-power beam width (HPBW) of 60° and a wider radiation profile. The prototype was tested against commercial helical, dipole, and patch antennas. It was found that the prototype antenna's maximum transmission distance was comparable to or exceeded other antennas, except the patch antenna, due to its HPBW of 75° and reasonably high gain.

Considering the feasible radio link capabilities, the next effort was to investigate network topologies for CubeSat constellations. Collaboration was formed with the Space Systems Group, in the Mechanical and Aerospace Engineering Department at the University of Florida, to generate possible satellite constellations. Then, the satellite constellations from Satellite Toolkit were imported to a network simulator. The simulation consisted of communications using UDP/IP from a sun-synchronous repeating ground track constellation and a flower (elliptical repeating ground track) constellation. Simulation results showed that a sun-synchronous constellation with a repeating ground track provided longer access time, lower drop-ratio, and higher throughput.

Then, the dissertation addresses the challenge of intermittent connectivity for CubeSat networks through evaluation of the store and forward DTN Bundle Protocol (BP) for CubeSat satellite environments in the S-band. An experiment was designed to simulate the physical and data link layers of the communication protocol stack, and

emulate DTN, on virtual machines. Results demonstrated that adding BP to UDP/IP delivered 100 times more data from an orbiting payload to ground station compared to just UDP/IP.

To increase downlink data rates over long propagation times, DTN protocols were also evaluated over an optical channel emulation testbed. The testbed emulated propagation delay, asynchronous channel rates, and bit errors. The testbed emulated an optical transceiver relaying data between virtualized ground stations. Free-space optical bit errors varied as a function of time for both uplink and downlink channels. The maximum goodput (i.e. effective throughput) was measured for various DTN configurations while increasing the round trip times. Results showed that DTN bundle protocol goodput surpassed UDP/IP after 200ms of round trip time. The following provides an overview of the issues to internetworking space communications.

Radio Frequency Allocation

The majority of CubeSat projects in last eight years and that are planned to launch in the next two years still utilize transceivers and beacons that continue to downlink in the UHF band and utilize the AX.25 protocol with no inter-satellite links. Some CubeSats do plan to operate at frequencies up to 5.8 GHz and as low as 145 MHz.

In the near future, CubeSat programs could use higher frequencies in either the C-Band or X-Band and further reduce the size and mass of the transceiver and the antenna and gain additional bandwidth to support payloads that have a significant data downlink requirement. The designers would have to consider the utility of additional bandwidth and decreased size and weight against increased power requirements to close the link with the ground station as the energy-per-bit is decreased for the same

power consumption. As CubeSat power generation systems become more effective and the satellites achieve three-axis stability, higher operating frequencies become increasingly feasible while permitting miniaturized components and increased antenna gain. However, designers must account for the increased gain trade off with beamwidth. In which case, a highly precise pointing mechanism in the attitude determination control systems would be required [3]. With this new technology, an update to the AX.25 protocol is needed to further CubeSat mission capability such as mission needing clusters or constellations.

Application and Orbital Properties

Many conclusions can be drawn from analyzing large ISL systems application and orbital properties. First, a satellite application dictates a constellation's specific topology design. Geosynchronous systems such as TDRSS, Luch, DRTS, and ESA's relay have made use of linking LEO spacecraft such as Hubble, Space Shuttle, and ISS. CubeSats have always been launched for LEOs. As an LEO spacecraft, a CubeSat may pass over a ground station ten minutes at a time, three times in 24 hours at best. Linking to a geosynchronous point could provide nearly round-the-clock communication coverage.

In some cases, multiple candidate constellation types may be appropriate for the same task. Traditionally, to minimize deployment cost, the constellation types were selected to minimize the number of satellites given the constellation's coverage requirements. However, when designing constellations of multiple satellites that communicate over inter-satellite links, the constellation's network performance (i.e., the quality of the inter-satellite links) should become a criterion for constellation selection.

Delay Tolerant Networking

The bundle DTN protocol can complement the CCSDS deep space protocol and near-earth orbit commercial protocol stack layers as a layer that runs over both. The DTN layer may do what the Internet did in the 1970s by layering over all networks by ignoring the properties of the lower layer and sitting on top. The DMC used IP on the ground stations and in orbit. However, due to asymmetric links and different transport layer like UDP should replace TCP. As a NASA DTN-IP debate progressed, it was decided that there was room for both IP and DTN, and it is a matter of finding the best solution for the application [4].

Some problems with DTN bundling have been learned from experiments in space. First, if the communication system does not implement the bundle protocol security, there is no error detection at all. If you do, everything is considered an attack. This is a reliability issue because there is no error detection, and reusing security to give reliability is not ideal.

In addition, it was decided that every spacecraft should have a clock and that clock should resynchronize with the ground. However, bundles are set with expiring times meaning when the bundle arrives with a missed set clock the bundle agent would drop the entire bundle. If the spacecraft has an incorrect time, the protocol cannot make a time correction because the spacecraft's clock is in the past. In addition, every bundle agent is expected to know current UTC. This has limits in space because the spacecraft clocks drift with temperature. Synchronization is a problem. Bundles can be dropped when expiring.

Optical Communications

Larger optical communication payloads have been tested and proved in Artemis, Spot-4, Envisat, Adeos-II, OICETS, Kodama, DAICHI, and SDS-1 at extremely high wireless data rates. However, the technology has not fit in the CubeSat form [5]. However, NASA has recently invested in demonstration of an optical modem for CubeSats. Since optical links can be easily blocked by cloud coverage that makes ISLs in the optical spectrum even more important. Thus, an optical intersatellite links between LEO spacecraft and larger geosynchronous relays and delay tolerant network protocols could mitigate cloud blockage.

High-rate communications could revolutionize space science and exploration. Space laser communications could enable applications that need high bandwidth such as imaging. The data throughput goal for the optical downlink is 100 Mbps. To compare, data from the Mars orbiter is sent back at a rate of six Mbps.

Organization

Chapter 2 provides a trade study of launched intersatellite linking systems, CubeSat communications, and internetworking satellite issues. Chapter 3 describes the design and analysis of deployable antenna mechanisms to address the challenge that most commercial off-the-shelf (COTS) directional antennas exceed a CubeSat's volume constraints. Chapter 4 describes an investigation into network topologies for CubeSat constellations. In Chapter 5, Delay Tolerant Networking (DTN) communication protocol suite is applied to overcome challenged communication environments of intermittent connectivity, and high bit error rate. In Chapter 6, DTN performance is evaluated for optical communications for long transmission delays. Chapter 7 concludes the dissertation with an examination of future work.

Communication Sub-systems for Intersatellite Linked Systems and CubeSat Missions

Intersatellite links or crosslinks provide connectivity between two or more satellites, thus eliminating the need for intermediate ground stations when sending data. Intersatellite links have been considered for satellite constellation missions involving earth observation and communications. Historically, a large distributed satellite system has needed an extremely high budget. However, the advent of the successful CubeSat platform allows for small satellites of less than one kilogram. This low-mass pico-satellite class platform could provide financially feasible support for large platform satellite constellations. Chapter 2 surveys past and planned large intersatellite linking systems. Then, Chapter 2 chronicles CubeSat communication subsystems used historically and in the near future. Finally, it examines the history of inter-networking protocols in space and open have changed the communication philosophy from research issues with the goal of moving towards the next generation intersatellite-linking constellation supported by CubeSat platform satellites. The research issue of link budget margin is then addressed in the Chapter 3 by presenting a deployable antenna for CubeSats.

Enhancing CubeSat Communication Through Effective Antenna System Design

A CubeSat's low-rate communication capability limits functionality. As greater payload and instrumentation functions are sought, increased data rate is needed. Since most CubeSats currently transmit at a 437 MHz frequency, several directional antenna types were studied for a 2.45 GHz, megabit capacity transmissions. This higher frequency provides the bandwidth needed for increasing the data rate. A deployable antenna mechanism maybe needed because most directional antennas are bigger than

the CubeSat size constraints. From the study, a deployable hemispherical helical antenna prototype was built [3]. Transmission between two prototype antenna equipped transceivers at varying distances tested the helical performance. When comparing the prototype antenna's maximum transmission distance to the other commercial antennas, the prototype outperformed all commercial antennas, except the patch antenna. The root cause was due to the helical antenna's narrow beam width. Future work can be done in attaining a more accurate alignment with the satellite's directional antenna to the downlink of a terrestrial ground station. In terms of intersatellite linking, effective communication will be impacted by the network topology investigated in Chapter 4.

Topology Design and Performance Analysis for Networked LEO Satellites

Many planned earth observation satellite missions could improve spatial and temporal resolutions using distributed satellite platforms in constellation or cluster orbits instead of one, large satellite with many instruments. A network of satellites and ground stations also allows the most data to be efficiently downlinked [6]. However, to create a networking protocol for constellations and clusters of satellites several limitations need addressed, including distributed topology management, slow downlink data-rates, and single point-to-point communication. Since distributed satellite constellations exacerbate the severity of these limitations, thorough analysis of a constellation's network performance is required to ensure that task objectives are achievable. In Chapter 5, low earth orbit satellite constellations were compared the constellations' inter-satellite links and downlinks with respect to network metrics including access window time, drop-ratio, and throughput. These network metrics were evaluated using the Network Simulator. Though previous works have proposed satellite constellations for Earth observation; these constellations have not been analyzed for these network metrics. Results show

that a sun-synchronous constellation with a repeating ground track outperforms a flower constellation with respect to more access time, lower drop-ratio, and higher throughput. The simulations also determined the optimum media access control slot time and packet transmission intervals for long distance satellite links. Additionally, the throughput performance of user datagram was compared to bundle protocol by adding nodes with delay tolerant networking protocol implementations.

Simulating Delay Tolerant Networking for CubeSats

Delay Tolerant Networking (DTN) is a communication protocol suite seen as a solution for challenged environments with long transmission delay, intermittent connectivity, and high bit error rate. Most research for DTN communication relies on simulation to validate protocols since real-world deployments are often either very expensive or impossible in the case of space communications. Simulation results are sensitive to the level of realism, many of which do not implement a realistic radio model or networking stack. In addition, there is an issue of cross-paper comparability as researchers often create their own simulators to test DTN algorithms, so it is difficult to compare a new algorithm with existing ones unless you implement it on a variety of simulators. Thus, an experiment that could simulate the physical and data link layers, and emulate the higher, DTN layers, was designed for a satellite cluster application. When compared, DTN's data rate outperforms UDP/IP. Future work includes integrating the simulation platform with more DTN implementations such as the Interplanetary Overlay Network implementation with Licklider Transmission Protocol (LTP). As a result, this flexible and scalable simulation platform can aid in system integration by investigating networking performance of nodes in many environments, running native DTN software, as DTN network nodes.

Performance Comparison of DTN Protocols for High Delay Space-based Optical Channels

Chapter 6 presents a channel emulation testbed that supports multiple gigabit-bandwidth Ethernet clients and servers simultaneously. In addition to bandwidth capacity, this testbed modeled optical link properties with a channel emulation machine, and provided high flexibility by utilizing virtual LANs, and link aggregation with a gigabit switch. Client machines connected to the testbed through the gigabit switch, then the switch passed frames to a PC-based channel emulator. For experiments, the testbed emulated propagation delay, asynchronous channel rates, and bit errors. The testbed modeled free-space optical bit errors that varied as a function of time for both uplinks and downlinks. For experimenting, the testbed emulated an optical flight terminal relaying data between ground stations. The maximum goodput of various DTN configurations was measured while increasing the relay's round trip times. Results showed that for 200 ms to one second round trip times, BP/TCPCL/TCP transmitted higher goodput than BP/LTP/UDP, and TCP/IP. For round trip times longer than one second, BP/LTP/UDP transmitted higher goodput than BP/TCPCL/TCP, and TCP/IP.

CHAPTER 2

A TAXONOMY OF CUBESAT TRANSMITTERS, LINKS, AND LAUNCHES

Chapter 2 surveys the frequency, protocol, application, and orbit of satellite systems launched that featured a crosslink. With CubeSats seen as a viable platform for intersatellite communications, Chapter 2 details frequency allocation, protocol, application, orbit, and communication subsystems used for all CubeSats.

History of Intersatellite Linking Communication Sub-systems

Intersatellite links have appeared in certain large platform satellite constellations for over 30 years. Examples of these large communication relay constellations include TDRSS and Iridium. With a miniaturization of payload mass, the costs of manufacturing and launching CubeSats could make it viable to move from satellite to ground station links to a multi-hop network of orbiting nodes [2].

Table 2-1. History of Intersatellite Linking (ISL) Systems

Launch Year(s)	Satellite(s)	ISL Frequencies
1972-1978	OSCARs 6,7,8	146 MHz
1976	LES-8 and 9	36, 38 GHz
1983-2013	TDRSS	C, Ku, Ka
1985-1995	Luch	UHF, Ka
1994	ETS-6	2, 23, 32, GHz, Optical
1997	Navstar Block IIR	UHF
1997	Iridium	23 GHz
1998	Comets (ETS-7)	2 GHz
1994-2003	MilStar VII	60 GHz
1998	Spot-4	Optical
2001	Artemis	S, Ka, Optical
2002	Envisat	S-band
2002	Adeos-II	2 GHz, 26 GHz
2005	OICETS	Optical
2010	AEHF SV-1	60 GHz
2015	Iridium Next	23 GHz

Separating system tasks into many dispersed sensor nodes can increase overall constellation coverage, system survivability, and versatility. However, to understand the challenges involved in satellite networking, large-platform Intersatellite linking systems

were surveyed, and are summarized in Table 2-1. Details of the frequencies, protocols, applications, and orbits of the OSCARs, TDRSSS, and Iridium systems are presented.

1972, 74, 78 OSCARs 6,7,8

The Radio Amateur Satellite Corporation's (AMSAT's) first generation of Orbiting Satellites Carrying Amateur Radios (OSCARs) were launched in the early 1960s. Soon after, AMSAT began work on a second-generation known as OSCARs 6, 7, and 8. These amateur satellites were characterized by an Intersatellite repeater capability [7]. As shown in Figure 2-1, OSCAR 6 received at VHF frequency of 146 MHz and transmitted at 29.5 MHz with a repeater bandwidth of 100 kHz. OSCAR 7 had two repeaters. The first repeater was similar to OSCAR 6 except for a slight frequency change and increased output transmission power. OSCAR 7's other repeater received at UHF 432 MHz and transmitted at VHF, 146 MHz.

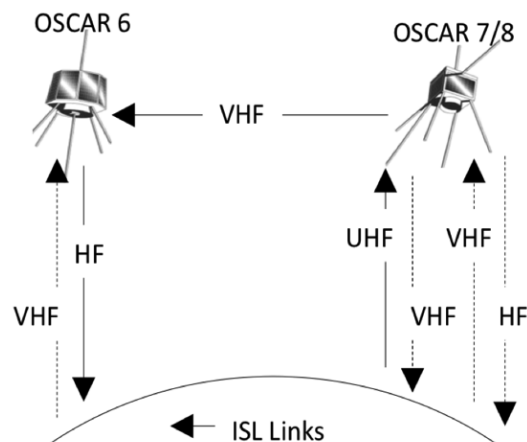


Figure 2-1. OSCARS 6 to OSCAR 7 and 8 carrier frequency bands

An on-board timer on OSCAR 7 automatically switched from one repeater to the other daily. In addition, satellite communication circuitry automatically switched one repeater on in a low-power mode when the battery was discharged to a certain point. As

a result, AMSAT demonstrated the first simple omnidirectional Intersatellite repeating system. AMSAT's OSCAR 7 remains semi-operational.

1976, LES-8 and LES-9

The Lincoln Laboratory experimental satellites 8 and 9, known as LES-8 and LES-9, demonstrated long-range digital transmissions in many bands between themselves and ground terminals. Once launched, LES-8 and LES-9 thrust 90° apart to a coplanar, inclined, circular, geosynchronous orbit [7].

Engineers at Lincoln desired crosslinks at a 60 GHz frequency band because of how oxygen molecules in the atmosphere absorb 60 GHz. This is much like how water absorbs the microwave band. However, with the technology in 1971, little test equipment was available beyond 40 GHz. As a result, LES-8 and LES-9 demonstrated crosslinks at the Ka-band (36 GHz and 38 GHz) [7]. 60 GHz crosslinks came later in MilSTAR constellation. Intersatellite antennas were either a 24 dB gain horn with 10° beamwidth or a more directional 42.6 dB gain dish with 1.15° beamwidth. The satellites were steerable +/-10° by a gimbaled flat plate [7].

As for a power budget, LES-8 and LES-9 had no solar cells or batteries. Instead, radioisotope thermoelectric generators (RTGs) powered the satellites. RTGs are the same sources that power the 1977 Voyager 1 and 2 spacecraft that have explored beyond the solar system.

An advantage of the Lincoln Lab's space communications program was that each LES was developed in the same facility. Lincoln Labs was able to experiment extensively for end-to-end communication testing including the ground terminals that Lincoln developed for the Air Force and Navy. The smooth testing following the launching of the satellites could be attributed to this.

Optical crosslink communication for LES-8 and LES- 9 was also considered. However, the idea was later dismissed in 1971 because it was considered infeasible with the current technology and beyond the resources of the project [7]. Optical crosslinks were accomplished with the Artemis satellite. So, one can conclude that LES-8 and LES-9 set the groundwork for many future satellite communication systems.

1983, TDRSS

Starting in 1983, NASA launched a series of geostationary satellites to relay low earth orbit (LEO), 300 km to 1,000 km, spacecraft communications. This Tracking and Data Relay Satellite System (TDRSS) satellite communication system, shown in Figure 2-2, increases the time window that spacecraft are in communication with ground terminals from ten-minute intervals to virtually anytime.

Another advantage of TDRSS is during atmospheric re-entry of manned missions. During re-entry of spacecraft during Mercury, Gemini, and Apollo missions, communication was lost due to the tremendous heating experienced by the craft and is termed "re-entry blackout". However, during the space shuttle missions, communication was not lost because TDRSS relayed transmission from the cooler conditioned antennas on top of the shuttle.

The most recent generation of TDRS provides downlinking data rates of 300 Mbps and 800 Mbps in Ku/Ka and S-bands respectively with five-meter diameter dishes [8]. In addition, a C-band phased array antenna can receive from five different spacecraft while transmitting to another location simultaneously [9].

TDRSS supports some of NASA's most famous missions. LEO spacecraft such as the space shuttle, Hubble telescope, LANDSAT, ISS all have their signals relayed from TDRSS to NASA ground terminals [10]. TDRSS has three main command and

control ground terminal locations at White Sands (WSGT) in New Mexico, Goddard (GRGT) in Greenbelt, Maryland, and Guam (STGT).

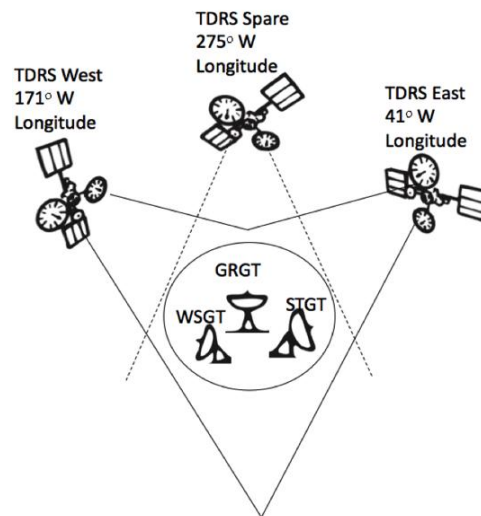


Figure 2-2. NASA's Tracking and Data Relay Satellite System (TDRSS). The three ground terminals are Goddard (GRGT), White Sands (WSGT), and Guam (STGT).

1997, Iridium

In 1997, Motorola Inc. built a constellation of LEO satellites called Iridium to provide global satellite phone coverage. The Iridium constellation operates with satellites that can intersatellite link, providing satellite calls and digital data to traverse through Iridium satellites instead of a cellular phone's network of base stations. As shown in Figure 2-3, the constellation includes 66 Iridium satellites and 6 spares orbiting in low earth orbit [11]. 7.5 km/second is the approximate orbital velocity for Iridium satellites, and LEO satellites in general. Each Iridium satellite can have up to four simultaneous crosslinks with a carrier frequency ranging from 13.18 GHz to 13.38 GHz, and a data rate of 10 Mbit/s [12]. Six satellites are contained in eleven orbital planes. Satellites in the same plane can link north and south or east and west with neighboring orbital planes. The polar orbits create two cross-seams. A cross-seam is where one

plane orbits south to north and the neighboring plane orbits north to south. As a result, the satellites only link with neighbor planes orbiting in the same direction due to a high Doppler shift. Iridium satellites orbit from pole to pole in approximate 100 minutes [13]. The Iridium satellite density at the North and South poles gives excellent coverage for research facilities.

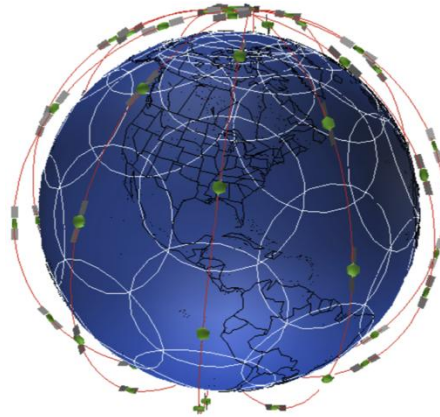


Figure 2-3. Iridium constellation generated by Satellite Constellation Visualization (SAVI)

History of CubeSat Communication Sub-systems

Starting in 1999, the CubeSat was an idea to standardize a platform for low-cost CubeSat missions and experiments, particularly at the university level. The CubeSat name is given from the pico-satellite's cube-like structure of 1 kg mass and 10 cm a side. A standardized structure allows CubeSats to share in launches with standard tubes and called Poly Picosatellite Orbital Deployers also known as P-PODs, shown in Figure 2-4. The P-POD Mk III has capacity for three 1U CubeSats however, since three 1U CubeSats are exactly the same size as one 3U CubeSat, and two 1U CubeSats are the same size as one 2U CubeSat, the P-POD can deploy 1U, 2U, or 3U CubeSats in any combination up to a maximum volume of 3U.

Over the past eight years, the CubeSat platform has flown successfully for many educational and scientific low earth orbiting satellite missions. To further CubeSat mission applications, a focus needs placed on inter-satellite networking that could improve the overall data rate and downlink time window of a CubeSat to ground terminal. Universities are developing a new robust, ad-hoc, long distance, high data rate protocol for CubeSats. However, in order to create this CubeSat protocol, a study of every inter-satellite linking mission and CubeSat mission communication system is presented in this report, categorized by launch date. A summary of this study is shown in Table 2-2 (for 2003-2005), in Table 2-3 (2006-2008), and in Table 2-4 (for 2009-2011). Special focus is put on the data downlinking transceiver and communication protocol used.

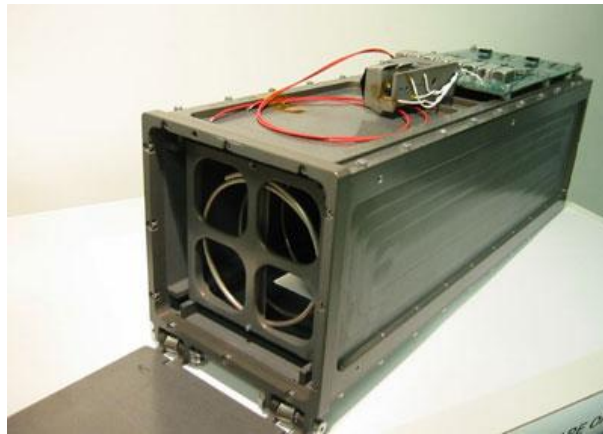


Figure 2-4. Poly Picosatellite Orbital Deployer or P-POD from Cal Poly holds three 1U CubeSats. Photo is courtesy of Bryan Klofas [15].

2003, Eurockot LV, Pletsak, Russia

Coordinated by University of Toronto's Space Flight Group in 2003, the first Nanosatellite Launch System (NLS) provides a low-cost launch service for CubeSat [14]. The first CubeSats were launched into a polar sun-synchronous orbit at 810 km from Plesetsk, Russia [15]. The launch integration effort for three 1U satellites was

known as Nanosatellite Launch service 1 (NLS1). NSL1 CubeSats included AAUSat from Aalborg, DTUSat from Denmark, and CanX-1 from University of Toronto. The three satellites were launched from the first P-POD. NLS2 launched QuakeSat, a 3U CubeSat from Stanford University, which took up an entire second P-POD. Also on the Eurockot, two Tokyo Pico-satellite Orbital Deployers or T-PODs, were deployed CUTE-I from Tokyo Tech. and XI from Tokyo University.

The two communication system characteristics that were common among the six CubeSats were the use of the AX.25 Link Layer Protocol specification and the frequency range from 432-438 MHz for their beacons and downlink. The only deviation from the AX.25 protocol was the CanX-1 satellite due to a proprietary nature of the information it was collecting and the addition of the PacSat layer to QuakeSat-1. However, CanX-1 Radio contact was never established. The choice of the AX.25 protocol allows amateur radio operators worldwide to collect information from the satellites and enhance the ground station effort. By utilizing the amateur radio community, the satellites could potentially transmit more data, thus provide more utility, if the data could be effectively re-assembled by the ground station [16].

QuakeSat-1 used PacSat Protocol Suite, part of the UoSAT and Microsat File transfer protocols. The suite defines file transfer protocols for use by LEO Packet Radio Satellites (PACSATs). Published in 1990, was the first digital store and forward packet protocol used for CubeSats [17].

Generally, CubeSats use Commercial Off-the-Shelf (COTS) radios, and modify them for use in space. The only published exception from the use of COTS equipment on this mission was the Japanese CubeSat XI-IV, which used a transceiver and beacon

that was developed within the University. However, the first three development versions of the satellite integrated a COTS transceiver until the flight model was ready [18].

Table 2-2. History of CubeSat Data Transmitters by Launch Date 2003-2005

Launch Date & Location	CubeSat Name	Size	Baud Rate	Frequencies
June 30, 2003	AAU1	1U	9600	437.475 MHz
Eurockot	DTUsat-1	1U	2400	437.475 MHz
Plesetsk Cosmodrome	CanX-1	1U	1200	437.880 MHz
Russia	Cute-1 (CO-55)	1U	1200	437.470 MHz
	QuakeSat-1	3U	9600	436.675 MHz
	XI-IV (CO-57)	1U	1200	437.490 MHz
October 27, 2005	XI-V (CO-58)	1U	1200	437.345 MHz
SSETI Express	NCube-2	1U	1200	437.505 MHz
Plesetsk Cosmodrome				2279.5 MHz
Russia	UWE-1	1U	1200/9600	437.505 MHz

As shown in Table 2-2, the results from the first batch of CubeSats have been mixed. The two Japanese satellites, Cute-1 and XI-IV, were successful, as they sent telemetry data until December 2008 and March 2009 respectively. The 3U QuakeSat-1 with PacSat can also be considered a success because it worked for more than seven times the design life of six months, and provided significant usable data. QuakeSat-1 also demonstrated the use of deployable solar panels for a higher maximum transmitting power. CanX-1 and DTUsat-1, however, never functioned on orbit and AAU1 had a significantly decreased lifetime, due to battery packaging problems, and a degraded communications system that led to only beacon packets transmitted for the life of the satellite [19]. The most notable characteristic for the launch is the use of dedicated beacon transmitters by the two Japanese satellites Cute-1 and XI-IV. These satellites also utilized dedicated antenna designs for the transceiver and the beacon, monopoles for Cute-1 and dipoles for XI-IV [18].

2005, SSeti Express, Pletsak, Russia

Upon the success of the first CubeSat launch more educational CubeSat projects were started. Sponsored by the European Space Agency Education office, a micro-satellite called SSeti Express was launched in 2005 in a sun-synchronous polar low earth orbit [15]. Although the SSETI Express microsatellite failed almost immediately after launch, three CubeSats including XI-V (University of Tokyo), NCube-2 (University of Oslo and others), and UWE-1 (University of Wurzburg) were successfully launched [15]. These CubeSats, known as NLS3, brought together many universities across Europe, and educated hundreds of students [14].

As shown in Table 2-2, the CubeSats all operated in the Amateur Band and used the amateur AX.25 protocol to simplify the ground station operation and increase the number of worldwide downlink points. Though the types of transceivers are unknown, it is known that NCube-2 and UWE-1 utilized commercial transceivers and the XI-V satellite family (identical to XI-IV in every aspect) continued to use the transceivers and beacons developed within the University of Tokyo.

NCube-2, with a UHF and experimental S-band transmitter, never transmitted to its ground station, and was declared dead on orbit. XI-V and UWE-1 both functioned as intended and utilized monopole antennas for their communications subsystem. Again, the Japanese Satellite XI-V used a separate beacon to provide redundancy [15].

UWE-1 was designed to test TCP/IP protocols in space and the effects of low bandwidth, long path delays, and dropped packets [18]. The main processor performed all the TNC functions, packing all the data into the AX.25 frame [20]. It then sent these frames using a 6-pack protocol (similar to KISS) to the TNC. This allowed the main processor to control the Data Link Layer settings of the radio, improving system

performance. Numerous other ground stations around the world received these beacons and forwarded the data on to the university [18], but the CubeSat stopped functioning in November 2005, about three weeks after launch [15].

Table 2-3. History of CubeSat Data Transmitters by Launch Date 2006-2008

Launch Date & Location	CubeSat Name	Size	Baud Rate	Frequencies
February 21, 2006 MV8	Cute-1.7 APD	2U	1200/9600	437.505 MHz
July 26, 2006	Ion	2U	1200	437.505 MHz
DNEPR-1 Failed	Sacred	1U	1200	467.870 MHz
Baikonur Cosmodrome	KuteSat Pathfinder	1U	1200	437.385 MHz
Kazakstan	Ice Cube-1	1U	9600	437.305 MHz
	Ice Cube-2	1U	9600	437.425 MHz
	Rincon1	1U	1200	436.870 MHz
	SEEDS	1U	1200	437.485 MHz
	HauSat1	1U	1200	437.465 MHz
	Ncube1	1U	9600	437.305 MHz
	Merope	1U	1200	145.980 MHz
	AeroCube-1	1U	9600	902-928 MHz
	CP1	1U	15	436.845 MHz
	CP2	1U	1200	437.425 MHz
	Mea Huaka (Voyager)	1U	1200	437.405 MHz
December 16, 2006 Wallops, US	GeneSat-1	3U	15 kbaud	2.4 GHz
April 17, 2007	CSTB1	1U	1200	400.0375 MHz
DNEPR-2	AeroCube-2	1U	38.4 kbaud	902-920 MHz
Baikonur Cosmodrome	CP4	1U	1200	437.325 MHz
Kazakstan	Libertab-1	1U	1200	437.405 MHz
	CAPE1	1U	9600	436.245 MHz
	CP3	1U	1200	437.845 MHz
	MAST	3U	15 kbaud	2.4 GHz
April 28, 2008	Delfi-C3 (DO-64)	3U	40 kbaud	435.55 MHz
Satish Dhawan, India	Seeds-2 (CO-66)	1U	1200	437.485 MHz
	CanX-2	3U	16-256 kbaud	2.2 GHz
	AAUSAT-II	1U	1200	437.425 MHz
	Cute 1.7 APD-II	3U	9600	437.475 MHz
	Compass-1	1U	1200	437.405 MHz
August 3, 2008	PREsat	3U	1200	437.845 MHz
Omelek, Marshall	NanoSail-D	3U	15 kbaud	437.505 MHz
Islands Falcon 1 Fail				

2006, M-V-8, Uchinoura, Japan

Single CubeSat launches have been rare. However, Cute 1.7 + APD was a 2U CubeSat that was individually launched by Tokyo Tech from Japan and tested an upgrade to the Tokyo Pico-satellite Orbital Deployer (T-POD) that could accommodate 2U CubeSats. CUTE 1.7 + APD had a highly elliptical orbit from 250-280 km in perigee height and 750 km in apogee height. This resulted in a short lifetime of over two months [15]. The CubeSat demonstrated use of a tether, an avalanche photodiode, a magnetic torque attitude control system (ACS), and a computer from a personal digital assistant. During the flight time, CUTE 1.7 successfully transmitted about one megabyte of mission information to the ground station [15].

CUTE-1.7+APD downlinked at 430 MHz and received uplink at 1200 MHz. The L-band uplink was used as a public personal message box for space. The message box performed at store and forward experiment by transmitting stored messages for periods to amateur radio ground stations. CUTE-1.7+APD was also known as OSCAR-56 because it was an operational digital repeater [15].

2006, Minotaur-1, NASA Wallops, VA, USA

GeneSat-1 was a joint project between NASA and academia that sought to evaluate biological payload experiments in a CubeSat [21]. GeneSat-1 was launched by a Minotaur rocket to a circular orbit with an altitude of 415 km [15]. The primary telemetry, tracking, and command link for GeneSat was a Microhard MHX-2400 frequency hopping spread spectrum Radio operating at 2.44 GHz and transmitted through a patch antenna. GeneSat also used a beacon operating in the Amateur Band from 432-438 MHz to serve as a risk reduction for the Microhard Radio, and to provide amateur radio operators the opportunity to collect satellite information [21]. GeneSat1

was downloaded the 500 KB required for its primary mission and continues to transmit beacon information [18]. NASAs GeneSat-1 3U platform bus, seen in Figure 2-5, has been used in NASAs later PharmaSat, PREsat, O/OREOS, and NanoSail-D CubeSats.

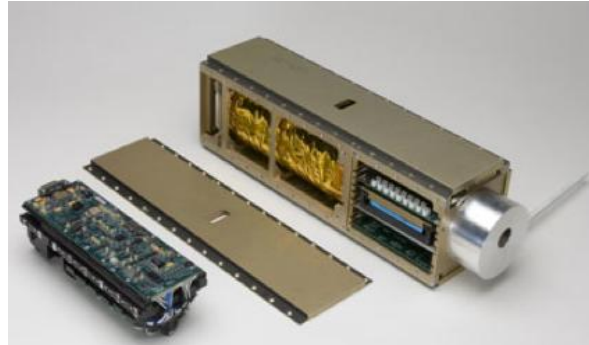


Figure 2-5. GeneSat-1's 3U bus at NASA Ames Research Center has been replicated for future NASA CubeSats such as PharmaSat, PREsat, and NanoSail-D1/2. Photo is courtesy of Bryan Klofas [18].

2007, DNEPR-2, Baikonur, Kazakhstan

Unlike the first Dnepr launch, this DNEPR LV launch successfully deployed three P-PODs in space, dropping the satellites in a polar sun-synchronous. The fourth major CubeSat launch contained several unique elements as Boeing entered the fray of the CubeSat community with their entry of CTSB-1 and Tethers Unlimited launched three tethered CubeSats known as MAST. Aerospace Corporation also launched their second CubeSat and California Polytechnic Institute launched CP3 and CP4, their third and fourth CubeSat [22].

Satellites CTSB-1, AeroCube-2, and MAST utilized proprietary packet protocols for their missions. These missions were also unique in their use of the frequency spectrum and the licensing requirements associated. CTSB-1 used an experimental license that operated at 400 MHz, AeroCube-2 used ISM license between 902-928 MHz

[22], and MAST used the same transceiver as GeneSat-1, a Microhard MHX2400 transceiver operating at 2.4 GHz.

The remaining satellites in the flight continued to use the AX.25 protocol along with frequencies within the UHF band. These satellites have had a mixed rate of success. Essentially all of the satellites have established communications with the ground. The exceptions were CAPE, which was integrated with a non-functioning receiver due to time constraints and Libertad-1, which had a non-functioning ground station when it was launched and the university personnel were not able to complete repairs in time to communicate with the satellite [18]. The remainder had down linked telemetry from several hundred kilobytes to several megabytes over the course of a year [18].

2008, PSLV-C9, Satish Dhawan Space, India

The first launch of multiple CubeSat outside of the former Soviet Union was supported as Nanosatellite Launch Systems 4 and 5 (NLS4 and NLS5) [14]. The launch consisted of satellites from Canada, Europe, and Japan. The satellites were launched using an eXperimental Push Out Deployer (X-POD), a launch system built for custom dimensions from pico-satellite to large nanosatellite classes. Notably, Delfi-C3 from Delft University of Technology in Holland was deployed to test wireless link data transfer within the satellite and new thin film solar cells.

As shown in Table 2-3, every satellite within the fifth batch used the 432-438 MHz portion of the AM band for part of its communications subsystem. CanX-2 also used the 2.390-2.450 GHz portion of the Amateur band for an additional downlink using a modified AX.25 protocol that the design team named the Nanosatellite Protocol (NSP) [23]. CanX-2 used two patch antennas for the S-band transceiver and a quad-canted

turnstile antenna for the downlink in the 70 cm band [15]. CUTE 1.7 + APD II used the Amateur frequencies available from 1240-1300 MHz for its uplink and maintained the AX.25 protocol for its transmissions [23]. According to AMSAT, all the satellites launched in the fifth batch are still in operation and transmitting to the ground station, except Compass-1, which is semi-operational as of September 2011.

2009, Minotaur-1, Wallops, MD

Table 2-4 shows the CubeSat launches for 2009-2011. In 2009, the US launched a batch of CubeSats, which included NASAs second CubeSat, Pharmasat-1, along with Aerocube-3, Hawksat-1, and CP6 from California Poly-technic State University aboard a Minotaur-1 rocket [18]. Pharmasats subsystems were mostly the same as Genesat. NASA developed the Microsatellite Free Flyer program that will leverage a standard subsystem baseline in 1U of the cube and allow various payload configurations in the remaining 2U of the satellite. Aerocube-3 and Hawksat-1 did not publish information regarding their projects. CP6 has also limited information on the subsystems available to the public. However, based on the published frequencies and available computer assisted drawings of CP6 and Hawksat-1, it can be assumed that the communications systems use a half-wave dipole.

2009, STS-127 Space Shuttle Endeavor

Space Shuttle Endeavor's STS-127 mission segment known as LONESTAR-1 launched the "DRAGONSAT payload" a satellite (12.7 cm) known as BEVO-1 and a CubeSat called AggieSat-2 in July 2009. BEVO-1, however was never operational upon launch [18]. However, AggieSat2 experimented with a unique dual-GPS system known as DRAGON by Johnson Space Center. To send commands to the CubeSat and poll responses, Aggiesat-2 utilized software called client [18]. The Client application allowed

AggieSat 2 lab member to conduct mission operations such as downlinking temperature, battery, and DRAGON GPS data during the CubeSat overpass. Ninety-nine contacts were made by Texas A&M students, and another thirty contacts were sent in by amateur radio operators around the world during AggieSat-2's 230 days of orbit.

Table 2-4. History of CubeSat Data Transmitters by Launch Date 2009-2011

Launch Date & Location	CubeSat Name	Size	Baud Rate	Frequencies
May 19, 2009	PharmaSat	3U	15 kbaud	2.4 GHz
Minotaur-1	CP6	1U	1200	437 MHz
Wallops, US	HawkSat-I	3U	1200	425 MHz
	AeroCube-3	1U	Proprietary	900 MHz
July 30, 2009 STS-127	AggieSat-2	1U	1200	436.25 MHz
September 23, 2009	SwissCube	1U	1200	436.25 MHz
ISI Launch 01	ITUpSat1	1U	19.2 kbaud	43 MHz
India	UWE-2	1U	9600	437.485 MHz
	BeeSat	1U	4800/9600	437.465 MHz
May 20, 2010	Hayato (K-Sat)	1U	10 kbit/1Mbit	13.275 GHz
Japanese H-IIA	Waseda-Sat1	1U	9600	437.485 MHz
Japan	Negai	1U	1200	427.305 MHz
July 12, 2010	TISat-1	1U	110 WPM	437.305 MHz
PSLV-C15 India	StudSat	1U	9600	437.505 MHz
November 20, 2010	O/OREOS	3U	1200	437.305 MHz
STP-S26	RAX1	3U	9600	2.4 GHz
Kodiak, Alaska	NanoSail-D2	3U	1200	437.275 MHz
August 12, 2010	Perseus (4)	1.5U	Proprietary	Proprietary
Falcon 9-002	QbX (2)	3U	Proprietary	Proprietary
Cape Canaveral	SMDC-ONE	3U	Proprietary	Proprietary
	Mayflower	3U	1200	437.6 MHz
March 4, 2011	E1P	1U	1200	437.505 MHz
Tarus XL Failure	Hermes	1U	56.2 kbit	2.4 GHz
Vandenberg, CA	KySat	3U	1200	436.79 MHz
Satish Dhawan, India	JUGNU	3U	20 WPM	437.275 MHz
October 28, 2011	DICE-1/2	1.5U	1.5 Mbit	PI
Elana3	M-Cubed	1U	9600	437.485 MHz
Vandenberg, CA	RAX-2	3U	9600	2.4 GHz
	E1P-2	1U	1200	437.505 MHz
	AubieSat-1	1U	20 WPM	437.475 MHz

2009, ISI Launch 01, Satish Dawan Space Center, India

The Innovative Solutions In Launch 01 (ISI Launch 01) was the second launch from India. All CubeSats were of a 1U platform. SwissCube-1 used an experimental sensor to detect nightglow, earth's atmospheric emission of light that allows earth sky to never be completely dark, even after the reflected light from stars is removed. When SwissCube was first launched in September of 2009, satellite rotated too fast to take pictures. This was likely a consequence from the deployment of its antennas.

Researchers and students had no stable picture until the CubeSat stabilized on its own, which could take up to a year. There would be no guarantee that SwissCube could resist solar radiation and extreme temperature changes for longer than four months. However, in November 2010, after more than one year of waiting, the rotation of Swisscube eased, but SwissCube's functions had deteriorated. Hoping to restore the defaults, a system restart was planned. To restart, the team purposely drained the batteries by communicating with the CubeSat as long as possible upon an overpass. To the relief of the team, the system rebooted as expected and all functions were restored. In early 2011, using three electro-magnets, the team aligned SwissCube with the magnetic field and stabilized the CubeSat. SwissCube-1 took its first visible light picture during February 2011 and its first airglow picture during March 2011 shown in Figure 2-6 [24].

ITUpSAT1 (Istanbul Technical University PicoSatellite-1) mission goals were to demonstrate a passive CubeSat stabilization system and to downlink an image from a CMOS camera payload of 640x480 pixel resolution. A MicroHard MHX-425 transceiver was used for ITUpSAT1's communications sub-system. The MHX-425 transmits with one watt of power, configurable frequency hopping patterns, and a high sensitivity of -

115 dBm for receiving uplinks. ITUpSAT1 downlinked 19200 baud from the MHX-425 transceiver.

BeeSat-1 or Berlin Experimental and Educational Satellite 1, was built to demonstrate reaction wheels for attitude control and had a 640x480 pixel resolution camera. BeeSat was also equipped with a sprightly 60 MHz ARM-7 CPU, 2 Gb of SRAM, 4 Mb of store and forward telemetry memory queue and 16 Mb of flash memory [25].

The University of Wurzburg's Experimental satellite 2 (UWE-2) experimented with optimization for internetworking specifications to work in long propagation environments. All CubeSats were launched in to a sun synchronous, slightly elliptical orbit. The orbit has a 98-minute period of revolution. As of January 2011, SwissCube, BeeSat are still operational [18].



Figure 2-6. Nightglow picture from Swisscube. In low resolution, the picture shows airglow, a phenomenon luminescent, resulting from the formation of oxygen molecules O_2 , following separation by solar radiation. Photo is courtesy of Muriel Noca, EPFL Space Center [24].

2010, Japanese H-IIA F17, Tanegashima Space Center

Three 1U CubeSats Hayato, Waseda-SAT2, and Nagai were launched from the Yoshinobu Launch Complex at the Tanegashima Space Centre [18]. Two JAXA

Picosatellite Deployers or J-PODs released the CubeSats. Hayato and Negai shared a J-POD and Waseda-SAT was stored in a second J-POD. These satellites were launched in extremely low earth orbits of 300 km. Hayato contained a dielectric patch antenna for X-band communication and a pantograph space boom of 60 cm for space weather study of water vapor in the earth's atmosphere. Waseda-SAT2 was launched for earth observation and to test the use of extending paddles to provide attitude control. Negai was launched to perform an orbital Field-programmable gate array (FPGA) experiment. Nagai also returned images of the Earth.

2010, PSLV-C15, Satish Dhawan Space Centre

Supported as NLS6 [16], TISat and the Indian StudSat were launched into a 700 km sun-synchronous orbit [18]. The Studsat took images of earth's surface with a resolution of 90 meters. TISat-1 is experimented with the durability of exposed thin bonding wires and printed circuit board tracks. TISat-1 also developed in-house all baseband modulation schemes in firmware. Both TIsat-1 and StudSat are currently operating with amateur radio stations around the world

2010, STP-S26, Kodiak, Alaska

Launched from Kodiak Island Launch Complex, Alaska, RAX-1, O/OREOS, and NanoSail-D2 piggybacked on the Formation Autonomy Spacecraft with Thrust, Relay, Attitude, and Cross-link (FASTRAC).

The goal of the RAX-1 mission was to measure the plasma instabilities in the lower polar thermosphere that can amount magnetic field-aligned irregularities. Magnetic field-aligned irregularities in between 80 to 400 km can jam wireless communication signals of spacecraft downlinking to ground stations. RAX1 used a standard 2-watt UHF AX.25 communication sub-system to downlink data back to a

ground terminal in Michigan. Due to an aberration in RAX-1's solar panels, the CubeSat was unable to generate power after several months. However, during operation RAX-1 was able to downlink bistatic radar measurements never attempted at this altitude.

RAX-2 is being rebuilt for a NASA's Elana 3 launch. NASA's O/OREOS or Organism Organic Exposure to Orbital Stresses is a 3U CubeSat built to study microorganisms' health in space. Microorganism's health data was downlinked using UHF AX.25, so amateur radio stations around the world can receive and forward the packets to NASA website for study [26]. O/OREOS also send data through S-band transmission to NASA over a Microhard MHX-2420 transceiver [26]. O/OREOS was built with a mechanism of two aluminum plates that separated with a spring. This mechanism increases the device's volume by 60%, creating drag, to slowly deorbit the CubeSat.

Last, NASA's NanoSail-D2 30-square-meter sail has was unfurled and the satellite has lowered its altitude. In fact, NanoSail-D2's can be seen at observatories. Orbital attitude, sail direction of pointing, and solar activity all affects NanoSail-D2's ascent rate [27].

2010, Falcon 9-002, Cape Canaveral, FL

The last CubeSat launch for 2010 was done from a Falcon-9 rocket. Eight CubeSats piggybacked on an evaluation mission for SpaceX's Dragon C1 capsule to provide supplies to the ISS in the future.

Two P-PODs contained four 1.5U satellites from Los Alamos National Laboratory called Perseus. The only known communication includes successful testing of two-way communication, three-way communication (two ground stations and a satellite), and collection of telemetry [18].

Another two P-PODs contained two 3U QbX satellite from the NRO's Colony-1 program [18]. No information has been further published. One P-POD housed an experimental US Army communications 3U CubeSat known as SMDC-ONE. SMDC-ONE uses a UHF transceiver to poll sensors to collect data and relay the data to ground terminals [18].

The last P-POD housed 3U CubeSat known as Mayflower-Caerus. The CubeSat was built as a joint mission. One unit of the 3U of the CubeSat was Caerus from the University of Southern California. The other two units made up Mayflower from Novaworks of Northrop Grumman. The CubeSat has experimental propulsion systems and eight extra solar panels that deploy as two arrays, the CubeSat downlinks using the UHF band to amateur radio stations [18]. The CubeSat re-entered the atmosphere during December of 2010.

2011, Jugnu, Satish Dawan Space Center, India

Built by the Indian Institute of Technology Kanpur, Jugnu is a 3U CubeSat to demonstrate Micro Imaging System, a near infrared camera to monitor crop irrigation. For tracking the Jugnu uses a GPS receiver and inertial measurement unit (IMU). The CubeSat uses UHF CW signals and radio amateurs have been reporting receiving signals since late 2011.

2011, Elana, Vandenberg AFB, CA

On October 28th, 2011 three P-PODs carrying six piggybacked CubeSats on a Delta II launched successfully from Vandenberg Air force Base. All three P-PODs deployed successfully and ground operators around the world are tracking the CubeSats. As of the evening of October 2011, signals from Aubiesat-1, E1P-2 and RAX-2 had been received.

The Dynamic Ionosphere CubeSat Experiment (DICE) is a CubeSat mission with the goal to map the geomagnetic SED (Storm Enhanced Density) plasma bulge and plume formations in earth's ionosphere. DICE-1 and DICE-2 are two identical 1.5U spin-stabilized spacecraft that downlink to a USRL ground station. DICEs' L-3 Cadet transceiver can downlink at a maximum data rate of 1.5 Mbps [28].

Michigan Multipurpose Minisat's (M-Cubed) objective is to obtain a mid-resolution photo from a single Cube-Sat [18]. Data and commands are transmitted using the AstroDev Lithium 1 radio. A dedicated receiver will operate at all times, while the dedicated transmitter will be operated only to send a beacon signal or transmit picture data. Both receiver and transmitter are the same component, AstroDev Lithium 1s, hard-wired to their independent tasks [18].

Radio Aurora Explorer 2 (or RAX2) is a redesigned version of RAX1 to overcome to power problems that occurred in the first launch. Significant changes from RAX1 to RAX2 include seven solar cells per panel instead of eight allowing increased photo-diodes as sun-sensors. The communication sub-system is identical to RAX1.

Explorer 1 Prime's Geiger-Mueller Counter sent telemetry using a custom KISS protocol. Any TNC capable of operating in KISS mode can decode the beacon. The connection of a KISS TNC to the E1P Telemetry Decoder is accomplished via a serial connection. The transmitter used onboard the spacecraft has a crystal oscillator that suffers from extreme temperature fluctuations. This means that general purpose Doppler correction software used by many amateurs is unsuitable for tuning the E1P downlink. The unstable oscillator causes the downlink frequency to shift from the center frequency up to +/-12 kHz. To allow the receiving station to decode a packet, a tracking

tone at 2200 Hz is downlinked. The EP1 staff then wrote software for ICOM radios to automatically tune to the wavering frequency using ICOM's CI-V connection.

AubieSat-1 is a 1U CubeSat built by students of the Auburn University to demonstrate gamma ray monitoring instruments during thunderstorms at high altitudes. AubieSat-1 uses CW to downlink all data [18].

CHAPTER 3

ENHANCING CUBESAT COMMUNICATION THROUGH EFFECTIVE ANTENNA SYSTEM DESIGN

Currently, a CubeSat's downlink transmission of pictures and videos back to a terrestrial ground station is slow. The US allocates 420-450 MHz to amateur bands while 2.4-2.5 GHz is an ISM band [29]. Thus, raising the frequency from a low frequency (437 MHz) to a higher frequency (2.45 GHz) would yield 70 MHz more in bandwidth for high-speed transmission of multimedia.

Since most antenna types cannot easily fit within the CubeSat dimensions, deployable antenna types and array configurations were designed. A trade study among antenna types such as the Yagi-Uda, log-periodic, helical, dish, and patch was done to consider how feasible a deployable configuration is. Numerical Electromagnetic Code (NEC-2) and Sonnet Microwave Studio provided simulations for each antenna type's radiation profile and overall gain.

Based on simulation results, two deployable hemispherical helical antenna prototypes were built. The helical prototypes were then tested by transmitting data between two of them at varying distances. The helical prototype results were then compared to different commercial antenna types including dipole, patch, and helical antennas. Results from the experiment showed that the deployable helical outperformed all commercial antennas, except the patch. This study of antenna types for CubeSats revealed that the patch antenna would provide the most signal stability and highest signal strength. The deployable helical antenna provided more directionality, and increased gain. However, high antenna directionality requires a need to consistently align the transmitter and receiver. Future work can be done on how to attain a more accurate alignment with a CubeSat's deployable helical antenna and ground station.

CubeSat Antenna Trade Study

To downlink data, most CubeSats currently use deployable dipole antennas.

These dipole antennas have an efficiency of 25%, radiate power in a wider cone, and have a very low gain [30]. These characteristics result in high signal to noise ratio (SNR) making dipoles easily interfered with and power inefficient.

Table 3-1. Typical values for CubeSat Transmission Parameters

Variable	Value
P_{rx} = Power Received	-120 dBW
P_{tx} = Power Transmitted	0 dBW
G_{tx} = Gain of Transmitting Antenna	5 dBi
G_{rx} = Gain of Receiving Antenna	10 dBi
d = Distance	350 km
λ = Wavelength	0.69 meters for 437 MHz

Depending on factors including ground station sensitivity, data rate, frequency, distance, and receiver antenna gain a CubeSat dipole antenna may need a transmit power of at least one watt. Typical values of wavelength (λ), frequency (f), distance (d), receiver gain (G_{rx}), and CubeSat transmitter gain (G_{tx}) are shown in Table 3-1. The typical distance of a low earth orbit (LEO) satellite is 350 km [15].

$$\begin{aligned}
 P_{rx} &= P_{tx} + G_{tx} + G_{rx} - 20 \log \frac{4\pi \times d}{\lambda} & (3-1) \\
 &= 0 \text{ dB} + 5 \text{ dB} + 10 \text{ dB} - 20 \log \frac{4\pi \times 350 \text{ km}}{.69 \text{ m}} \\
 &= -120 \text{ dB}
 \end{aligned}$$

CubeSat transmission antenna gain is typically 5 dBi and receiver antenna gain is 10 dBi. Since the transmission center frequency is 2.45 GHz, λ is set at .125 meters. From these values the power received on the ground is calculated by Friis' [29] formula. Friis' Equation 3-1 yields a received power of -120 when just looking at the path loss a

transmitted one watt CubeSat signal would incur. A high-speed 2.45 GHz transmitting antenna would have to be designed to match this power received.

Table 3-2 summarizes the baud rates for past 1U CubeSat transmitting radios. The 420-450 MHz amateur bands support a typical speed of 9600 bps [31]. 9600 bps is too low to transfer images or video. For transmitting images or video using CubeSats, a communication system should be capable of data transfer speeds of the order of Mbps. In the past 5 years, downlink data totals under 50 MB for 17 1U satellites [15]. Communications are a major bottleneck for CubeSat functionality.

Table 3-2. 1U CubeSat Transmitter Rates, Modulations, Frequency, and Power

CubeSat	CUTE1	AAU1	AERO2
Baud Rate	1200	9600	38.4 kbaud
Modulation	AFSK	GMSK	FSK
Frequency	437.47 MHz	437.475 MHz	902-928 MHz
Power Generation	350 mW	500 mW	2 Watts

Multiple Antenna System Design

As shown in Figure 3-1, The use of two orthogonal antennas can enhance CubeSat communication: a dipole unit for omni-directional communication and another antenna for downlink (satellite to ground) communication. The dipole antenna continues to operate at the low UHF-band (437 MHz) and is more practical for CubeSat data such as orientation and vital system information. This information, in single bits, does not require a high bitrate, but needs a consistent connection with the ground station. The high-speed downlink antenna would transmit at a higher 2.4-2.5 GHz band when sending megabit per second data capacity such as multimedia. The use of these two antennas saves overall transmission power by decreasing the transmission period. However, a directional antenna needs to be aligned toward the base station.

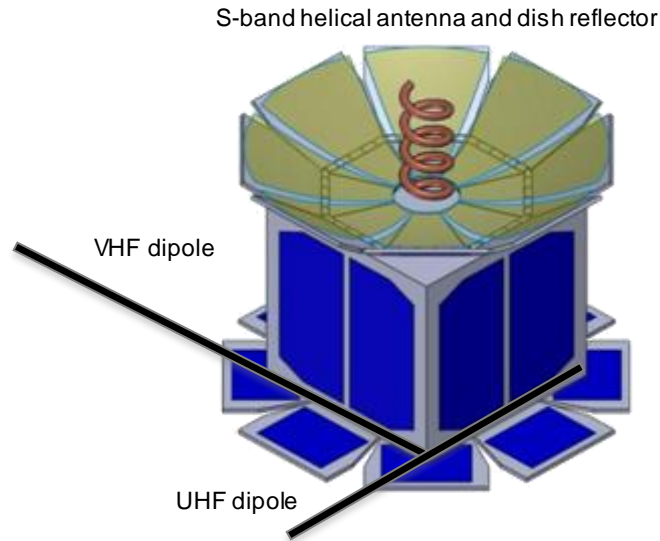


Figure 3-1. Multiple Antennas on CubeSat

Antenna Type Selection

The antenna type selection was based on other factors such as attitude disturbance, frequency capabilities, complexity, and feasibility. However, gain and size were the main factors. Directional antennas require more internal volume than dipoles when installing in a CubeSat. To combat the issue of size, directional antennas with an ability to deploy were studied. Deployment of a horn or dish reflectors would meet the gain needed. However, designing for the deployment of a dish or horn reflector in space is complex, so both of these options were eliminated.

Theoretical Results for Commercial Direction Antennas

Most CubeSats use dipole antennas, often made from the same metal as a tape-measurer, allowing the dipole to spring out during deployment. As CubeSats tumble, and spin the antennas jitter. All the CubeSats have used simple antennas like monopole or dipole because there is not much space for high gain, but complex, antennas like helical or dish. Also directional antennas like helical or dish require precision pointing ability because of their narrow beam width. However, with Rapid

Rotation and Precision Pointing mechanism, precision pointing is possible and thus high gain antennas were investigated.

Work was performed in the Wireless and Mobile Systems Laboratory at the University of Florida to simulate and test various antennas for down selection to the CubeSat format [5]. To select a directional antenna for high-speed downlink, the dish, horn, Yagi-Uda, log-periodic, and helical antenna types were simulated in NEC-2 to see the radiation profiles. Figures 3-2 to 3-4 show the radiation profiles and power density of 10 element, Yagi-Uda, log-periodic and helical antennas. All three antennas showed potential as a high-speed downlink antenna to be used in conjunction with a dipole antenna.



Figure 3-2. Radiation profile of a 10-element, Yagi-Uda antenna. Dimensions for 2400 MHz frequency: 14 dBi Simulated Gain, Beam width: 38°. Electrical Boom Length of 28.0 cm.

The Yagi-Uda antenna is a series of dipole antennas, giving it directionality. It is used commonly as a television antenna. Deploying a series of dipole antennas on a CubeSat is feasible. From the American Radio League (ARRL) book [29], Yagi-Uda

antenna gain is a function of the number of dipole elements, N , and can be expressed in Equation 3-2

$$\text{Gain}_{\text{yagi}} = 1.66 \times N \quad (3-2)$$

If a Yagi-Uda antenna has $N = 7$ dipole elements, the gain is 11.67 dBi. The spacing of the seven elements have to be $1/3$ of $\lambda = 12.5$ cm [29], making the Yagi-Uda antenna 29.2 cm long. The log-periodic antenna is worse than the Yagi-Uda because the log-periodic offers even less gain for its size [30].

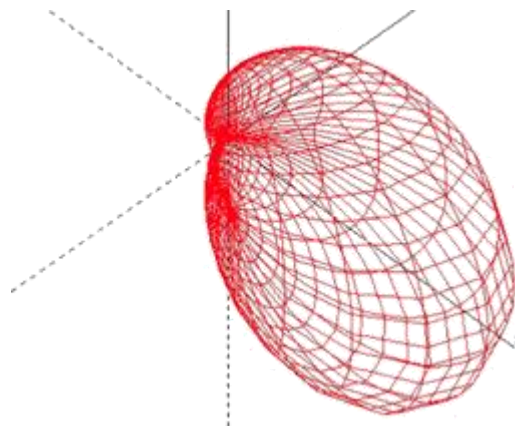


Figure 3-3. Radiation profile of a 10 element Log-periodic antenna: Dimensions for 2400 MHz, 12 dBi Simulated Gain, Beam width: 38° . Total length 59 cm. Total boom length 22 cm

A helical antenna is a conductive wire with spaced out wrapping resembling a spring. Table 3-3 shows the gain of a helical antenna as a function of the number of wraps or turns it makes. The gain is calculated by the Kraus [30] formula (3-7). After only five turns, the helical antenna's gain is 11.8 dBi, which surpasses that of a seven element Yagi-Uda. In addition, the spacing for each turn of a helical antenna is $1/4 \lambda$ instead of $1/3$ in the case of the Yagi-Uda [29]. A five-turn helical length calculates to 15.6 cm. So, a five-turn helical has more gain than a seven element Yagi-Uda and is half the length.

Table 3-3. Gain vs Dimensions for a 2.45 GHz Helical Antenna

Turn Number	Gain (dBi)	Length (cm)
3	9.55	9.37
4	10.8	12.5
5	11.8	15.6
6	12.6	18.8
7	13.2	21.9
8	13.8	25.0
9	14.3	28.1
10	14.7	31.3

In addition to the helical antenna having more gain in a compact stowed volume, the antenna is easy to tune, has circular polarity, and is simple to deploy. As a result, a 2.45 GHz deployable helical antenna prototype with dimension fitting a CubeSat was constructed for testing. The radiation profile for a 10 turn helical is shown in Figure 3-4.

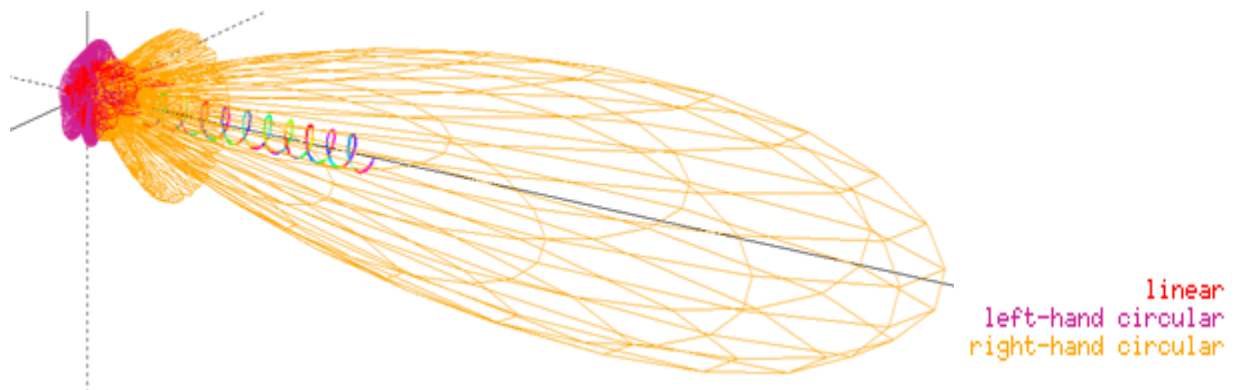


Figure 3-4. Radiation profile of a 10 turn Helical antenna: Dimensions for 2400 MHz, 14.8 dBi Simulated Gain, Beam width: 32.9°. Total length 31.25 cm

Since the helical is a spring like antenna, it deploys similar to a jack in the box. A burn-wire holding the helical would ignite once the CubeSat is in orbit. The compressed helical would expand into the antenna's proper shape. Nitinol, a conductive memory metal which can be shaped, compressed and expanded into its original shape could be used as the antenna material [32].

Deployable Helical Antenna Design

After consideration of helical, yagi, log-periodic, dish high gain antennas for CubeSats, yagi and log-periodic antennas were eliminated because of their construction (with many elements) despite their high gain. The many elements are difficult to deploy on a satellite and thus is not reliable. The helix is easy to tune and is compact in stowed volume relative to the antenna gain. Therefore, a helix has more gain in a simpler, compact package, when compared to a yagi. It is much easier to achieve circular polarity with a helix than a yagi. As a result, helical antennas were selected.

Prototype Dimensions

To obtain the correct dimensions for building a deployable helical and 3-3 to 3-7 were formulated. Equation 3-7, the theoretical gain for an axial helical antenna is known as the 'Kraus Model Formula' [30]. Equations for an axial helical antenna:

$$\text{Circumference of Helix } C = \lambda = 12.5 \text{ cm} \quad (3-3)$$

$$\text{Spacing Between Coils } S = \frac{\lambda}{4} = 3.125 \text{ cm} \quad (3-4)$$

$$\text{Helical Turn Radius } R = \frac{\lambda}{2\pi} = 2 \text{ cm} \quad (3-5)$$

$$\text{HPBW} = \frac{52^\circ}{\frac{C}{\lambda} \times \frac{N \times S}{\lambda}} = 39.3^\circ \quad (3-6)$$

$$G_h = 10.8 + 10 \log \left(\frac{C}{\lambda} \right)^2 \times N \times \frac{S}{\lambda} = 13.2 \text{ dBi} \quad (3-7)$$

The number of turns, N, was set to seven and applied to the above equations. The half-power beam width (HPBW) is calculated in Equation 3-6 as 39.3° for a cylindrical helical, but a hemispherical taper will be added to the prototype making the HPBW approximately 60° according to Cardoso [33].

Impedance Matching

Matching the impedance of a network to the impedance of a transmission line is crucial to an efficient wireless link. Equation 3-8 models the impedance of a helical antenna. If the Circumference and wavelength are the same, then the antenna impedance is 140Ω . However, there are solutions for a helical antenna to match standard 50Ω impedance, the characteristic impedance (Z) of a resulting helical transmission line.

To greatly reduce the impedance mismatch over a wide frequency band the last quarter turn of the prototype was tapered in. This technique can convert a 140Ω helix impedance down to 50Ω standard-coaxial-cable impedance at the feed point [34-35].

$$Z = 140 \times \frac{C}{\lambda} (\Omega) = 140 (\Omega) \quad (3-8)$$

Hemispherical Tapering

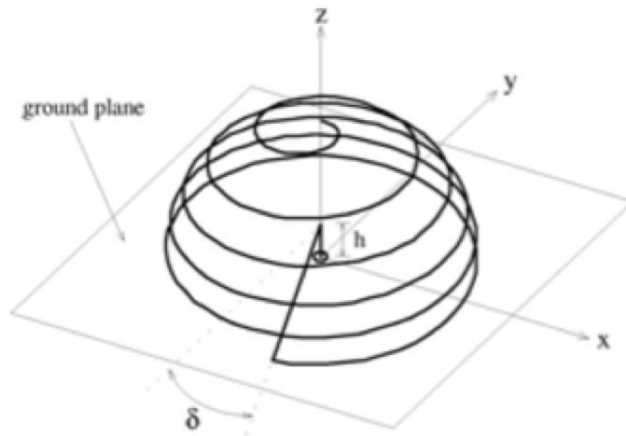


Figure 3-5. Detailed sketch of a hemispherical helical antenna

A semi-circular taper on an axial helical antenna is known as a hemispherical helical antenna, shown in Figure 3-5. Hemispherical helices are of great interest to a CubeSat because they can compress into a single plane. Thus, the tapered helical occupies little space in a CubeSat. This saves volume for the payload. This deployable

hemispherical helical prototype is pictured in Figure 3-6. The hemispherical taper increases the helical half-power beamwidth from that of a cylindrical helical of 39.2° to about 60° according to Cardoso [33].



Figure 3-6. Deployment for the prototype hemispherical helical, photo is courtesy of Paul Muri.

Performance

Table 3-4. Typical values for CubeSat Transmission Parameters

Variable	Value
P_{rx} = Power Received	-120 dBW
P_{tx} = Power Transmitted	0 dBW
G_{tx} = Gain of Transmitting Antenna	13.2 dBi
G_{rx} = Gain of Receiving Antenna	18 dBi
d = Distance	350 km
λ = Wavelength	0.125 meters for 2400 MHz

The power received from a seven-turn helical transmitting 350 km was calculated using Friis [30] Path Loss Equation 3-1. The radiation profile for a seven turn helical is shown in Figure 3-7. Table 3-4 shows how a 21-turn helical on a ground station could receive -120 dBW with a seven-turn helical antenna transmitting at 2.45 GHz. This is the same power received that a ground station typically sees with a 437 MHz dipole shown earlier in Table 3-1. However, now over three times more bandwidth is available

for a high bit rate transmission and the power transmitted has remained the same at one watt.

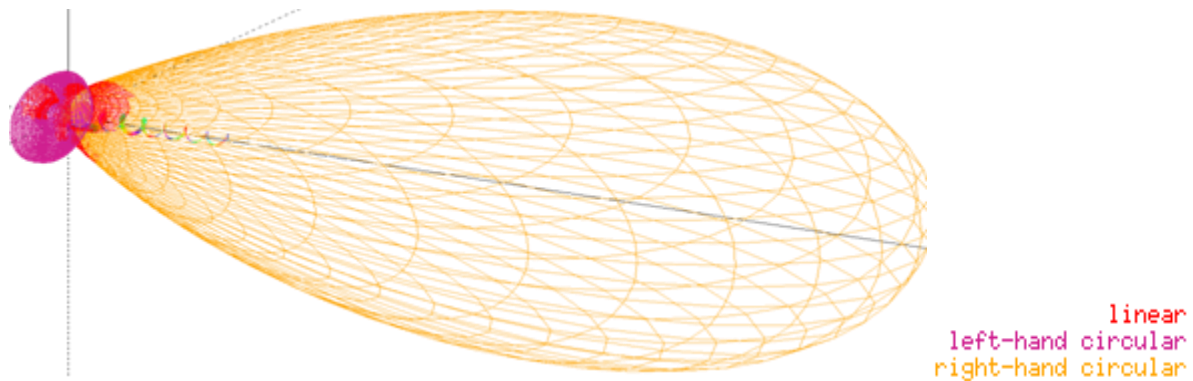


Figure 3-7. Radiation profile of a 7 turn tapered Helical antenna: Dimensions for 2400 MHz, 10 dBi Simulated Gain, Beam width: 60°. Total length 21.875 cm

Power Savings

The seven-turn helical antenna has a gain of 13.2 dBi. With respect to a half-wave dipole antenna gain of 2.15 dBi, the helical has a gain of 11.05 dBd. Equation 3-9 calculates the percentage power savings of the deployable helical when compared to a UHF half-wave dipole. Gain dB is the gain of helical antenna with respect to dipole antenna.

$$\text{Power Savings} = 100\% - \frac{100\%}{10^{\frac{\text{Gain}_{\text{dBi}}}{10}}} \quad (3-9)$$

Since the total power savings depends on the average period of each session, Table 3-5 analyzes the power savings that can be obtained using only a directional antenna. A high gain seven-turn helical has a 92.14% power savings when compared to the power a dipole needs to transmit.

As a result, a helical antenna with a higher gain can save 92% of the transmission power. However, the CubeSat needs to use an omnidirectional antenna for

bootstrapping a communication session. Some extra power is needed for this. The amount of exact overhead power drained depends on the average length of the session.

Table 3-5. Power Savings from Helical Antenna Gain

Turn Number	Gain (dBi)	Gain (dBd)	Power Savings (%)
4	10.5	8.35	85.37%
5	11.8	9.65	89.16%
6	12.6	10.45	90.98%
7	13.2	11.05	92.14%

Antenna Testing

A commercial patch, commercial helical, and prototype helical was characterized from single-port S11 s-parameter files from an Agilent N5230C Network Analyzer at the NASA Goddard Space Flight Center's Microwave Systems Branch Laboratory. The s1p file gave magnitude and phase for the antennas sweeping through frequencies of 2 GHz to 3 GHz. Screen shots of the SWRs for each antenna were taken. The real and imaginary components were converted from the magnitude and phase. Then, magnitude, and Smith charts were plotted from the s1p file using matlab.

Commercial Patch Antenna

The HyperLink 2.4 GHz 8 dBi Flat Patch Antenna Model: HG2409P is a directional patch that can affix to a 10x10cm face of a CubeSat. Table 3-6 describes the HG2409's specifications. Of the antennas tested, the commercial patch has the widest beam width at 75°, allowing more error in pointing accuracy.

Table 3-6. Commercial Patch Antenna Specifications

Parameter	Value
Type	Directional
Frequency Range	2400 – 2500 MHz
Gain	8 dBi at 2.437
Azimuth beam width	75°
Elevation beam width	65°
VSWR	<1.5:1
Impedance	50 Ω

Figure 3-8 shows how the patch was placed in an anechoic chamber, a room designed to completely absorb reflections of electromagnetic waves, at the NASA Goddard Space Flight Center's Microwave Systems Branch Laboratory. The chamber is insulated from exterior sources of noise. The combination of both aspects means they simulate a quiet open-space of infinite dimension, which is useful when exterior influences would otherwise give false results.

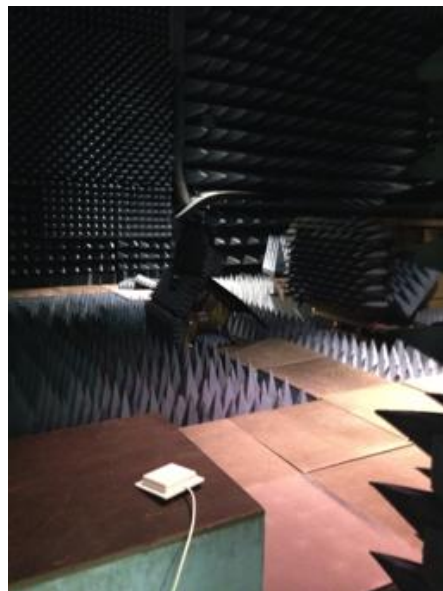


Figure 3-8. Patch antenna in an anechoic chamber at the NASA Goddard Space Flight Center's Microwave Systems Branch Laboratory. Photo is courtesy of Paul Muri.

Figure 3-9 shows the patch's standing wave ratio (SWR), and measure of how well a load is impedance-matched to a source. SWR is always expressed as a ratio with 1 in the denominator (2:1, 3:1, 10:1) It is a scalar measurement only (no angle), so although they reflect waves oppositely, a short circuit and an open circuit have the same VSWR value (∞ :1). A perfect impedance match corresponds to a SWR 1:1, but in practice, one never achieves it. Impedance matched means maximum power is

transferred from source to load. In general, SWR is permissible under 2:1 ratio of reflectance for the desired frequencies. Note that SWR for the commercial patch is acceptable between the wide frequency spectrum of 2.3 to 2.9 GHz

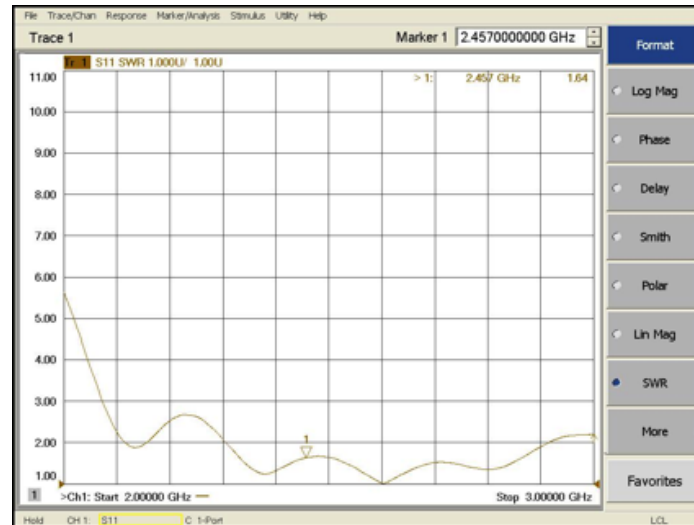


Figure 3-9. Patch SWR is under 2:1 from 2.3 to 2.9 GHz

S-parameters describe the input-output relationship between ports (or terminals) in an electrical system. S11 parameters would be the reflected power radio 1 is trying to deliver to antenna 1. S22 would be the reflected power radio 2 is attempting to deliver to antenna 2. In addition, S12 is the power from radio 2 that is delivered through antenna 1 to radio 1. Note that in general S-parameters are a function of frequency.

In practice, the most commonly quoted parameter in regards to antennas is S11. S11 represents how much power is reflected from the antenna, and hence is known as the reflection coefficient (sometimes written as γ : or return loss. If S11=0 dB, then all the power is reflected from the antenna and nothing is radiated. If S11=-10 dB, this implies that if 3 dB of power is delivered to the antenna, -7 dB is the reflected power. The remainder of the power was accepted by or delivered to the antenna. This accepted power is either radiated or absorbed as losses within the antenna. Since antennas are

typically designed to be low loss, ideally the majority of the power delivered to the antenna is radiated.

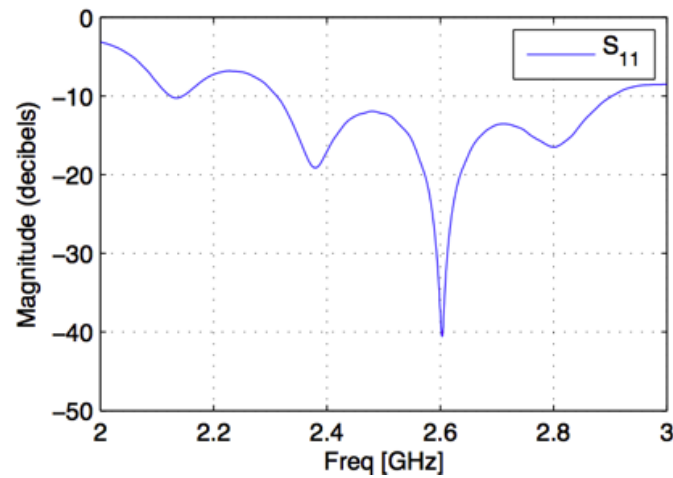


Figure 3-10. Patch Magnitude over frequency

Figure 3-10 shows the S_{11} Magnitude over frequency for the commercial patch. Similar to the proper SWR in Figure 3-09, the magnitude is under negative 10 dB from 2.3 GHz to 2.9 GHz, within the requirement needs for 2.4 GHz. The most reflect power occurs at 2.6 GHz, and the antenna has high resonant frequencies at 2.395 GHz and 2.805 GHz.

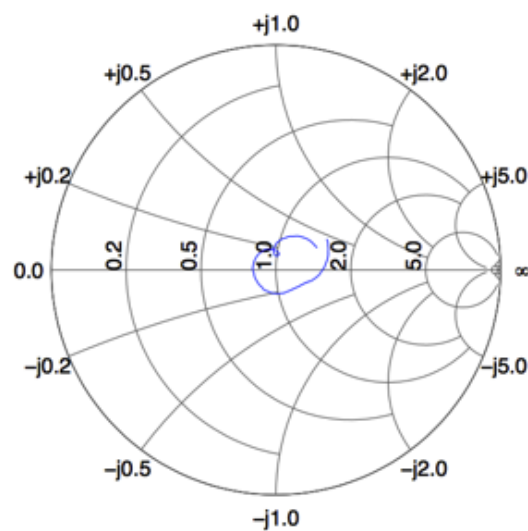


Figure 3-11. Patch Smith Chart

Figure 3-11 shows a Smith chart, a chart of the reflection coefficient to help visualize the impedance of an antenna as a function of frequency, for S11 parameters of the patch. The Smith chart in Figure 3-11 is drawn closest having a reflection coefficient of one, at the origin.

Commercial Helical Antenna

Table 3-7. Commercial Helical Antenna Specifications

Parameter	Value
Type	Directional
Frequency Range	2400 – 2500 MHz
Gain	9.6 dBi at 2.437
Polarization	Right Hand Circular
Connector	N female
Azimuth beam width	39.3°
Elevation beam width	50°
VSWR	<1.3:1
Impedance	50 Ω

The Luxul 10dBi, Circular Polarized X-WAV, Helical Antenna Model: XW-24-H10 is a directional patch that can affix to a 10x10cm face of a CubeSat. Table 3-7 describes the XW-24-H10's specifications. Of the antennas tested, the commercial helical had the narrowest beam width defined by Equation 3-6 at 39.3°, allowing more error in pointing accuracy.

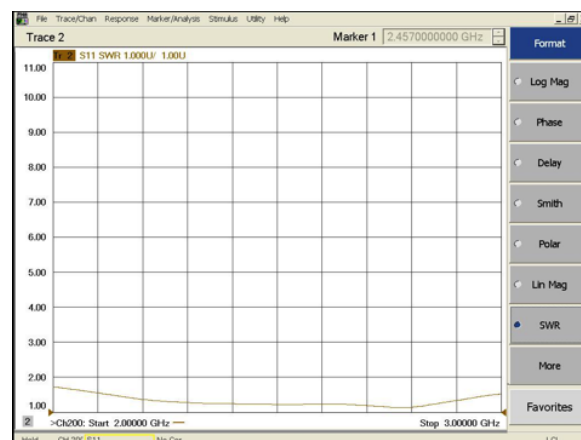


Figure 3-12. Luxul Helical SWR is under 1.5:1 from 2.1 to 3 GHz

Figure 3-12 shows the patch's SWR. SWR for the commercial helical is very well matched. The commercial helical's SWR is under 1.5:1 for the wide spectrum of 2.1 to 3 GHz, making the antenna easily within the requirements for impedance matching at 2.4 GHz.

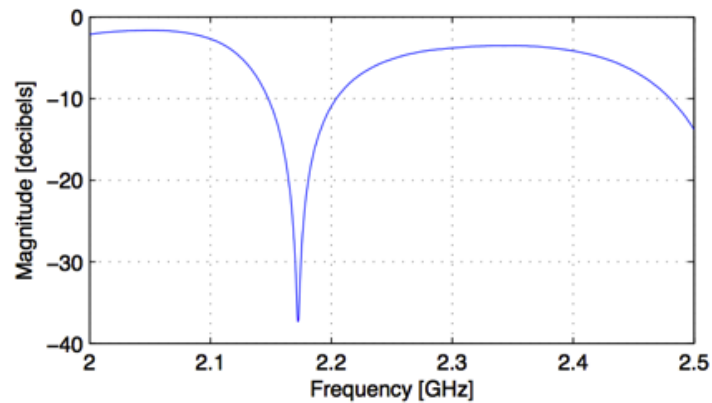


Figure 3-13. Luxul Helical Magnitude over frequency

Figure 3-13 shows the S11 Magnitude over frequency for the commercial helical. For the commercial helical antenna, the magnitude is under negative 10 dB between 2.15 GHz to 2.104 GHz, not within the requirement needs for 2.4 GHz. The most reflected power occurs at 2.182 GHz. Thus, the antenna was not expected to perform well for 2.4 GHz.

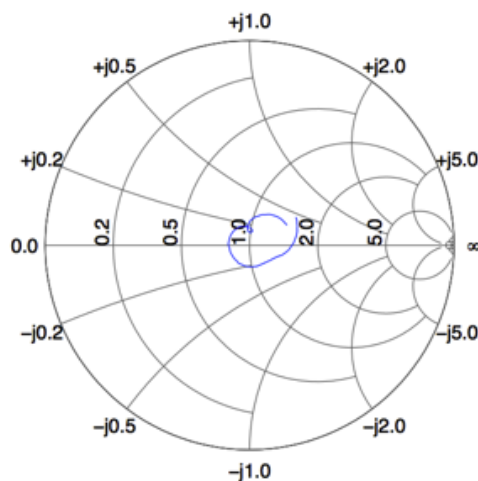


Figure 3-14. Luxul Helical Smith Chart

Figure 3-14 shows a Smith chart for the commercial helical antenna. The Smith plot is drawn closest having a reflection coefficient of one, at the origin. Thus, the 50 Ω impedance is well matched for the commercial helical.

Prototype Helical Antenna

A deployable prototype helical antenna that can be stored in a 10x10x0.5cm dimensions as a CubeSat payload was designed and built. Table 3-8 describes the prototype helical's specifications.

Table 3-8. Prototype Helical Antenna Specifications

Parameter	Value
Type	Directional
Frequency Range	2400 – 2500 MHz
Gain	13.2 dBi at 2.437
Polarization	Right Hand Circular
Connector	N female
Circumference (λ)	12.5 cm
Number of Turns (N)	7 turns
Spacing (S)	3.125 cm
Length of Antenna ($L=N \times S$)	21.8 cm
Length of Wire (L_{wire})	192 cm
Diameter (D)	3.98 cm
Radius (R)	2 cm
Conductor Thickness (T)	0.25 cm
Reflector Diameter	12.5 cm
Effective Aperture (A_e)	86.9°
Half-power beam width (uniform)	60°
VSWR	<1.3:1
Impedance	50 Ω

Of the antennas tested, the prototype helical had the median beam width allowing more error in pointing accuracy when compared to the commercial helical.

Figure 3-15 shows the prototype helical in the anechoic chamber at the NASA Goddard Space Flight Center's Microwave Systems Branch Laboratory.

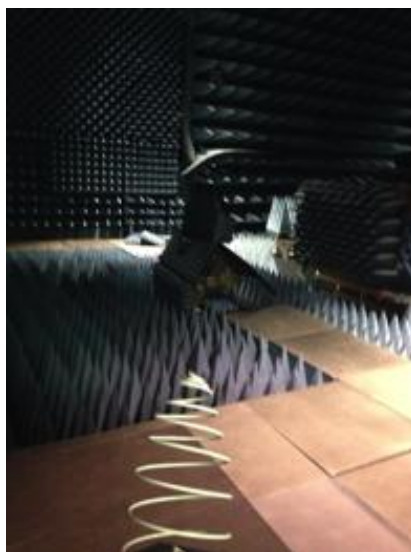


Figure 3-15. Prototype tapered in the anechoic chamber. Photo is courtesy of Paul Muri.

Figure 3-16 shows the prototype helical's SWR. SWR for the prototype helical is matched under 2:1 for the wide spectrum of 2.3 to 2.42 GHz, making the antenna within the requirements for impedance matching at 2.4 GHz.

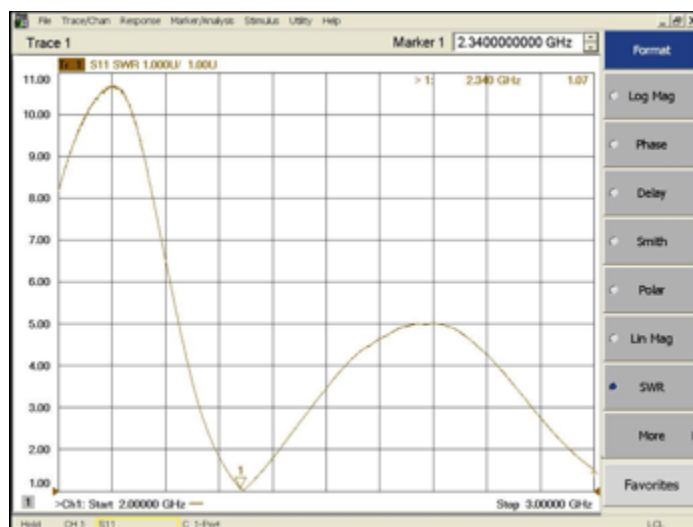


Figure 3-16. Prototype Helical is under 2:1 from 2.3 to 2.42 GHz

Figure 3-17 shows the S11 Magnitude over frequency for the prototype helical. For the prototype helical antenna, the magnitude is well under negative 10 dB the entire

frequency range of 2 GHz to 2.5 GHz, within the requirement needs for 2.4 GHz. The most reflected power occurs at 2.395 GHz.

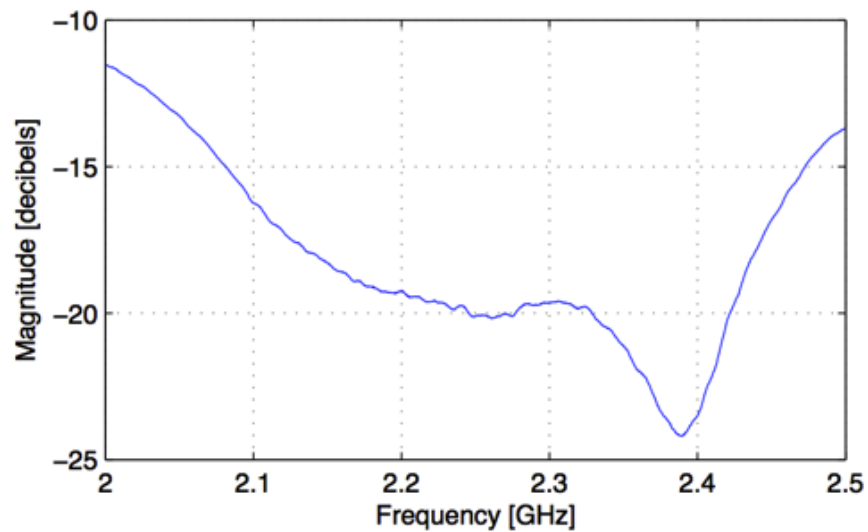


Figure 3-17. Prototype Helical Magnitude over frequency

Figure 3-18 shows a Smith chart for the prototype helical antenna. The Smith plot is drawn closest having a reflection coefficient of one, at the origin. Thus, the 50Ω impedance is well matched for the prototype helical.

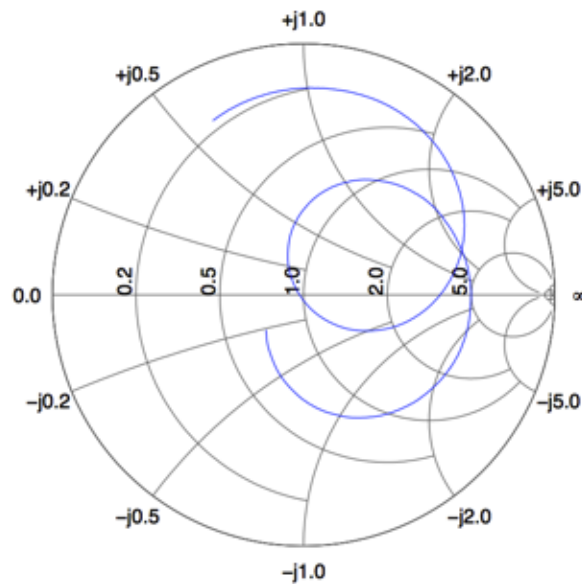


Figure 3-18. Prototype Helical Smith Chart

Experiment

To test the antenna theory, two deployable helical prototypes were built. The two antenna prototypes were field tested to find their maximum transmission range. Each antenna was connected to a 50 mW transceiver (XBee Pro 50mW Series 2.5 RPSMA). The two XBee Pro transceivers were used at both the downlink and uplink ends (with line of sight) to transmit and receive at 2.45 GHz. As the XBee transceivers continually sent bits to each other, measured received signal strength indication (RSSI) and recorded it. Five trials of transmissions were done at ten discrete distances varying from 0 meters to 900 meters with a 100 meter step. Figure 3-6 shows the deployable seven-turn hemispherical helical antenna prototype. Different commercial antennas were connected to the uplink and downlink transceivers for each trial. The commercial antennas used to benchmark the deployable helical prototype included a 3 dBi dipole, 5 dBi dipole, 9.6 dBi helical, and a 6 dBi patch.

Results

The commercial antennas' RSSI for each distance was recorded and compared with the custom made helical prototype model shown in Figure 3-19. Error bar indicate one standard deviation of error as a function to pointing and antenna HPBW. Pointing error increased with distance. Results on RSSI readings at various distances ranged from 0 to 900 meters. The 6dbi patch outperformed the other antennas in RSSI. Both the path, and the custom made deployable helical performed well having a maximum transmission range of 900 meters. The commercial 9.6 dBi helical did worse than expected only transmitting at a maximum distance of 500 meters. The two dipoles exhibited the worst performance and shortest range.

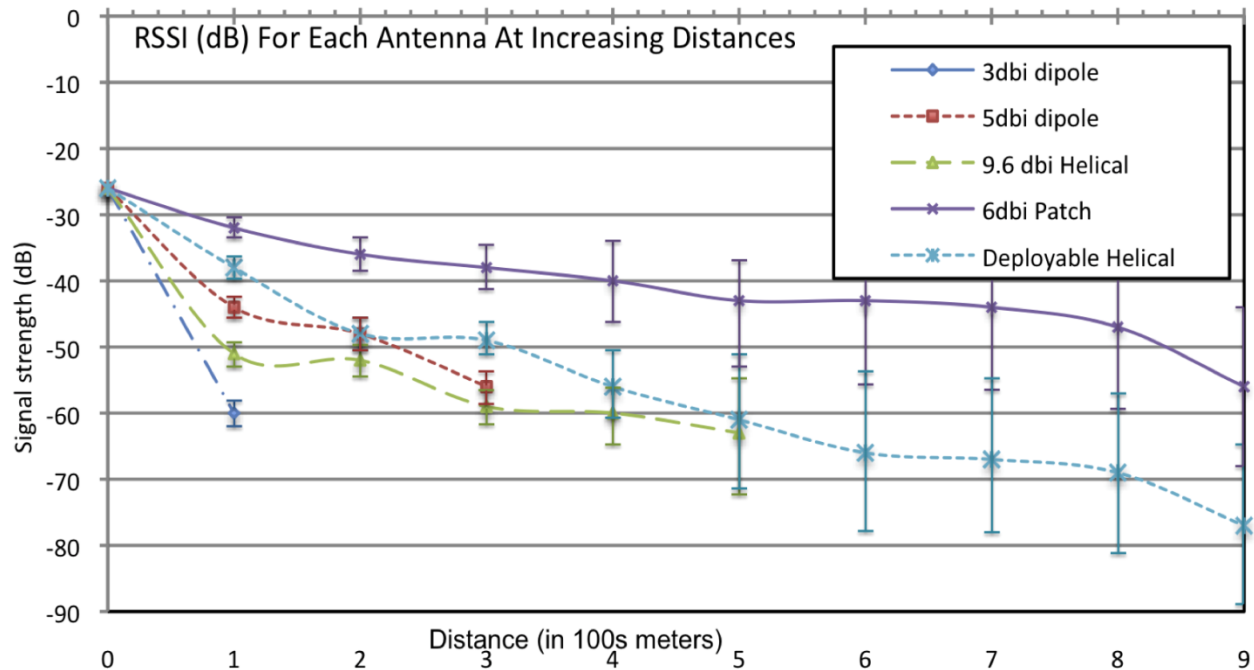


Figure 3-19. Deployable Helical Performance vs. Other Commercial Antennas

Summary

This investigation of candidate directional antenna types for CubeSats revealed that the patch antenna provides a stable, long-range transmission. The reason for this is that the 2.45 GHz commercial patch antenna had a high HPBW of 75° [36] along and reasonably high gain.

Since the tests did not carefully align the receiver to the transmitter, the directional cylindrical commercial helical antenna did not perform well having a maximum transmission distance of only 500 meters. This is because it had a more narrow beam width of 52°. The hemispherical tapered helical performed well because it had a tapered design, which leads to a higher HPBW of 60° [33] and wider radiation profile as mentioned earlier. Thus, a deployable taper helical antenna could provide increased gain with a wider beam width giving it a more stable transmission.

CHAPTER 4

TOPOLOGY DESIGN AND PERFORMANCE ANALYSIS FOR NETWORK LEO SATELLITES

Satellite constellations can be used to cover large areas of the earth surface. Examples include GPS for navigation, Rapid Eye for earth imaging, and Iridium for communications. A constellation's task dictates a specific design. In some cases, multiple candidate constellation types may be appropriate for the same task.

Sun-synchronous orbits have desirable features for remote sensing, Earth observation, and weather, since these orbits have near constant illumination angles. An equatorial repeating ground track orbit allows for approaching the ground station at an identical angle, and encountering the ground station more frequently than the sun-synchronous orbit at up to twelve times per day [37]. A repeating ground track satellite could act as a sink to relay the sun-synchronous CubeSats' observation data. However, all CubeSats would need propulsion capabilities to be placed in this constellation.

Table 4-1. Applications for Constellation Types

Constellation Type	Typical Task(s)
Sun-synchronous Repeating Ground Track (Polar Orbiting)	Earth observation Remote Sensing Communications
Flower (Elliptical Orbits with Sinking satellite in Circular Orbits)	Atmospheric/weather monitoring Experimental orbits for GPS Remote Sensing

With no propulsion required, satellite clusters allow for faster build times, simpler designs, more redundancy, higher resolutions, and multiple angles and times for observation. An example of this type of cluster is the QB50 program [38], which has called for a cluster of 50 CubeSats for a magnetosphere study. For QB50, CubeSats do not transmit through inter-satellite links. However, an inter-satellite networking capability

in such a mission could take advantage of the cluster topology for distributed processing and higher data relay to a ground station.

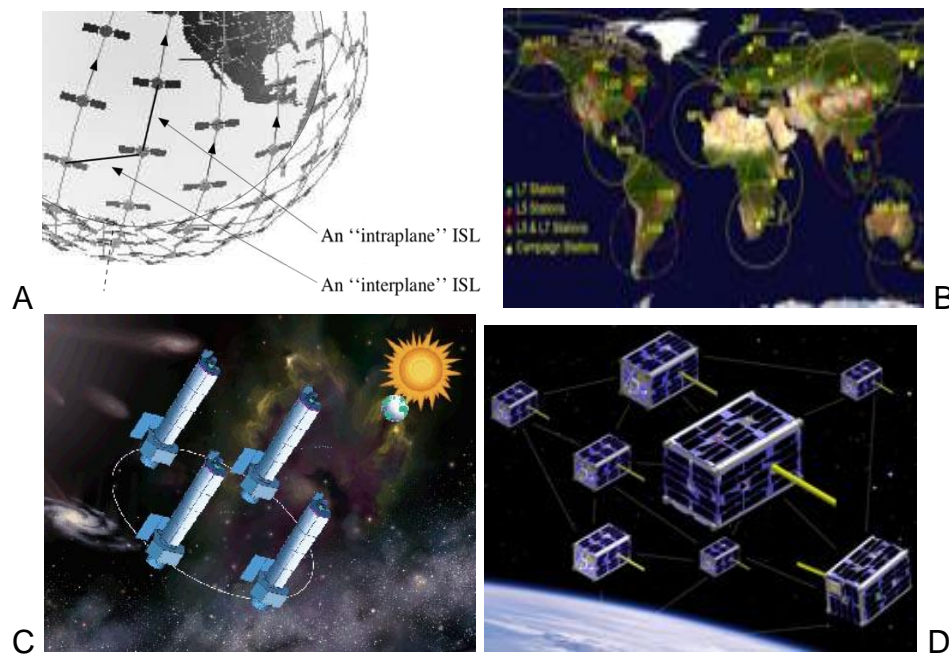


Figure 4-1. Series of distributed satellite constellations and clusters. A) Constellation with crosslinks, credit: [39] B) Constellation with Ground links, credit: [40] C) Formation Flying Cluster, credit: [40] D) Free Flying Cluster, credit: [41]

Traditionally, to minimize deployment cost, constellations were selected to minimize the number of satellites and yet maintain the coverage requirements. However, when designing constellations of multiple satellites communicating over inter-satellite links, the quality of the inter-satellite links becomes a significant criterion.

Therefore, in order to compare candidate constellation types for effective satellite mission design, the effect that the constellation's type has on the inter-satellite link network performance must be evaluated. Simulation is a widely used method for evaluating network performance, and satellite packages for are available for OMNeT++, OPNET, QualNet, and the Network Simulator (ns-2) [42]. However, most satellite simulations packages are for large satellites with more powerful transmitters than the

transmitters available on CubeSats. Some research focused on evaluating network performance of specific protocols for CubeSat constellations using network simulators [42], however these studies have focused on evaluating protocol performance or optimizing a single CubeSat constellation rather than comparing candidate constellation types using network performance as a criterion.

In Chapter 4, two constellations for low Earth orbit (LEO) CubeSat Earth observing missions were designed with SSRGT and flower constellation types in Satellite Toolkit (STK) [43]. A novel method for evaluating the network performance of arbitrary satellite constellations using the Network Simulator (ns-2), an established network simulator [39] was developed. This work was performed by a multi-disciplinary team of PhD students (Cason, Aerospace and Antoon, Computer Architecture) [6]. Team member contributions are noted within content.

Constellation Topology Design

To evaluate constellation type selection, Cason [6] designed two candidate constellations for a hypothetical Earth observing mission that monitors islands along the Sunda ocean trench for geological events. To reduce the complexity of the process, both constellations were designed to have the same spatial coverage, i.e., the area of the target a constellation observes [44].

Effects of limited CubeSat access to ground stations were mitigated by employing the data mule methodology [45]. The data mule methodology uses a set of source satellites to collect data, and a set of sink satellites to transport the data to a ground station. Sinks are larger, more powerful satellites with more access time to ground stations than source satellites, but less access time to target areas than source

satellites. In order to maintain relevancy between the two candidate constellations, each candidate constellation contained six sink satellites and nine source satellites.

Sun-synchronous Repeating Ground Track (SSRGT) Constellation

SSRGT orbits have desirable features for remote sensing and Earth observation applications, since these orbits have near constant illumination angles and approach targets with identical viewing angles up to twelve times per day [37]. These characteristics are amenable to Earth observation missions in both the visible and infrared spectrum. Satellite systems such as the LANDSAT program [46], and imaging and remote sensing satellites and constellations, such as Spot satellites [47], and RapidEye [48] leverage the SSRGT orbit.

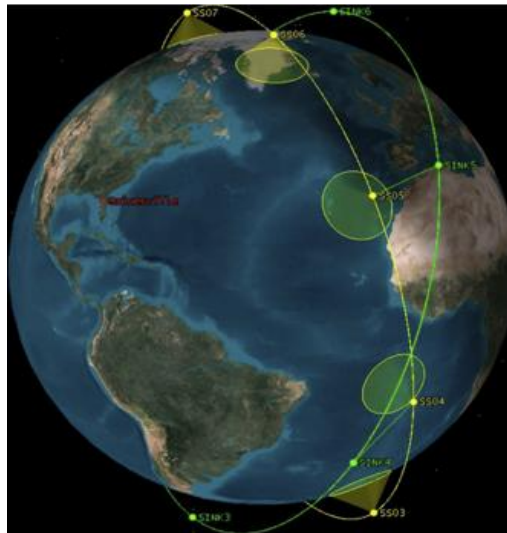


Figure 4-2. The SSRGT constellation shows the right-most lines near the North Pole crossing to the left-most near the South Pole represent the six sinks' orbiting path.

Cason in [6] configured the individual satellites in the SSRGT constellation to maximize the access time to the ground station. To maximize the access time between the source satellites and the target, the source satellites in the SSRGT constellation were distributed equally about a polar orbit at an altitude of 750 km. To maximize the

access time between the sink satellites and the ground station, the sink satellites were distributed equally about a circular orbit at a 70° inclination and an altitude of 750 km.

Figure 4-2 shows the SSRGT constellation scenario designed by Cason in [6].

Flower Constellation

The flower constellation is a repeating ground track orbit with an axis of symmetry that coincides with the spin axis of the Earth. Flower constellations are well suited for Earth observation because each source satellite in a flower constellation has the same orbit shape and all the satellite node lines are displaced equally along the equatorial plane [49]. Figure 4-3 shows the flower constellation design designed by Cason in [6].

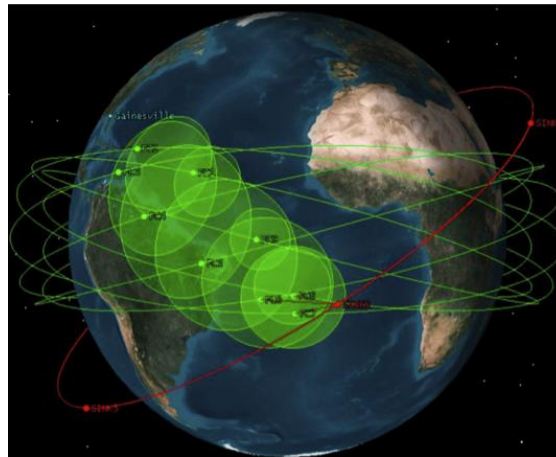


Figure 4-3. The flower constellation shows the bottom-most line on the left crossing to the top-most on right is the six sinks orbiting path.

Table 4-2 shows a comparison of the orbital parameters for the SSRGT and flower constellation designs. In order to maximize access time to the target, all nine source satellites are in an elliptical, near-equatorial LEO. In order to accommodate the data mule methodology and maximize sink satellite time to the ground station, the six

flower constellation sink satellites were distributed in a traditional circular 1598 km orbit with a 35° inclination.

Table 4-2. Orbital Parameters for SSRGT and Flower Constellation

Orbital Properties	SSRGT Sensing	SSRGT Sinks	Flower Sensing	Flower Sinks
Apogee Altitude	750 km	750 km	1598 km	1598 km
Perigee Altitude	750 km	750 km	686 km	1598 km
Inclination	97.3°	70°	165°	35°
Right Ascension of the Ascending Node	0°	0°	Satellites 1-9: 0, 40, 80, 120, 160, 200, 240, 280, 320°	0°
True Anomaly	Satellites 1-9: 0, 40, 80, 120, 160, 200, 240, 280, 320°	Satellites 1-6: 0, 60, 120, 180, 240, 300°	Satellites 1-9: 0, 54, 98, 134, 165, 195, 226, 262, 307°	Satellites 1-6: 0, 60, 120, 180, 240, 300°

Modeling LEO Satellite Constellations

Using Satellite Toolkit for Network Topology Design

The topology of the SSRGT and flower constellations was modeled using STK and evaluated the network performance using ns-2. In STK, both constellations were simulated over the same three month period, from January 12th, 2011 16:00 to April 12th, 2011 16:00. The simulation time step was one minute. STKs orbit wizard was used to define specific satellite parameters depending on orbit type. Not only were the same sink satellite orbits used for both simulations, but the same ground station (Gainesville, FL), and the same target area (Jakarta, Indonesia) were used as well to make the constellations as similar as possible.

Communication Range Threshold

Configuring the 802.11g-2007 standard can allow for acceptable long-range performance. The Friis transmission Equation 3-1 calculated that with 30dbm G_{tx} , 10db G_{rx} , and -116dbm sensitivity, nodes have a maximum range of 2,000 km. An assumption of 10db gain for an object in challenged networks was based on a study in Chapter 3 of how to point a CubeSat and deploy a directional antenna to the intended destination. The study applies whether the destination is a ground station or other satellite. Table 4-3 shows the transmission range parameters used.

Table 4-3. Parameters to Calculate Transmission Range

Variable	Value
P_{rx} = Power Received	-116 dBm
P_{tx} = Power Transmitted	30 dBm
G_{tx} = Gain of Transmitting Antenna	10 dB
G_{rx} = Gain of Receiving Antenna	10 dB
d = Distance	2000 km
λ = Wavelength	0.125 meters for 2400 MHz

To address the Doppler shift, Jakes propagation loss model, built into NS-3 can be used. The model's parameter for Doppler frequency shift, f_d [Hz] can be calculated from velocity V [m/s], transmission frequency f [Hz] and light speed c [m/s] in Equation 4-1.

$$F_d \text{ [Hz]} = \frac{V \text{ [m/s]}}{\lambda} = \frac{V \text{ [m/s]} \times f \text{ [Hz]}}{c \text{ [m/s]}} \quad (4-1)$$

In [50], a Doppler shift of 50 kHz is experienced in circular LEO satellite formations. Thus this orbit is within the 802.11 mobility specification with a maximum frequency variation of nominally ± 60 kHz [51]. A radio has this range of acceptable frequencies due to a Doppler shift when applying a phase lock loop. A phase-locked

receiver electronically tunes itself to track the Doppler-shift on the carrier frequency using a voltage-controlled oscillator.

Access time between the sensing satellites and the sink satellites determined the transmission window [6]. The access time is the period, in seconds, for two satellites to communicate with one another, given a range limit in kilometers. The range limit was a maximum of 2000 km, solved from the Path Loss Equation 3-1.

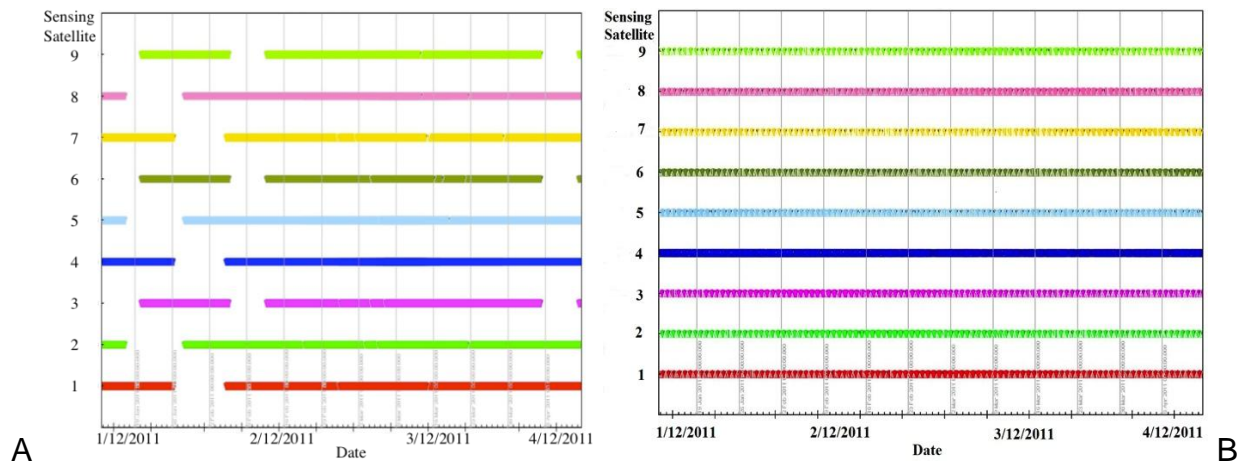


Figure 4-4. Access time windows for sensing to sink communication A) SSRGT constellation access time windows, note that windows are not continuous B) Flower constellation access time windows (credit: Cason [6])

STK's access tool calculated access start time, end time, duration in minutes, the number of accesses, and the maximum, and minimum durations. Figure 4-4 shows accesses between the nine sensing satellites and the six sink satellites for the SSRGT; these access time graphs were initially generated individually and then overlaid to show the overall transmission window that the sensing satellites have with the sink satellites. STK's access tool was used to generate the access time between each sink satellite and a ground station in Gainesville, FL. The access time for the ground station will determine how much time the sink satellites will have to down-link their data.

When establishing an Earth observing constellation mission, the mission's goal can either be to achieve high temporal or spatial coverage. Temporal resolution is the frequency with which an image can be captured. The more often a certain area is imaged then the better the temporal resolution will be. Spatial coverage is the amount of the Earth's surface the constellation covers over a given period [44]. The mission concentrated on temporal resolution over a certain area, Jakarta, Indonesia, which would be beneficial for a disaster monitoring situation such as a tsunami.

NS-2 to Evaluate Network Performance

The default ns-2 package contains multiple models for simulating satellite constellations and the obvious model choice for simulation is the ns-2 satellite model, which can simulate well-known constellations such as Iridium [14]. However, since this satellite model only supports circular orbits with un-slotted ALOHA-net as the link layer protocol, the ns-2 mobile node model with each satellite represented a satellite node.

The ns-2 mobile node model is robust and supports a wide range of protocols. However, since this model is most appropriate for terrestrial wireless networks, the ns-2 mobile node model was modified to simulate the constellations. Creating a mobile node simulation typically consists of plotting the nodes' movements as tool command language (Tcl) scripts, called scenarios, which are imported into ns-2. However, since ns-2 does not support three-dimensional (3D) positioning, a script to translate the 3D satellite movements into a scenario could not be written.

So, in order to integrate 3D movements into ns-2, ns-2's positioning system was modified. Unlike the other ns-2 simulation models, the ns-2 mobile node model is not easily modifiable and inhibits direct replacement of ns-2's positioning system. To overcome this restriction, new modules were written to replace the existing modules

interfaced with the positioning system. The new modules used an external database of satellite positions to provide node position information.

To use the STK constellation data in the new ns-2 modules, STK exported the SSRGT and flower constellations as comma separated value (CSV) files. The CSV files contained a position for each satellite at every minute. A Python script was written to translate the satellites' positions into a structured database and a C++ library was written to load, cache, and linearly interpolate the satellites' positions at times between the recorded minute positions. Using a C++ library, any ns-2 module that used node positions was replaced with a version that used STK-exported data.

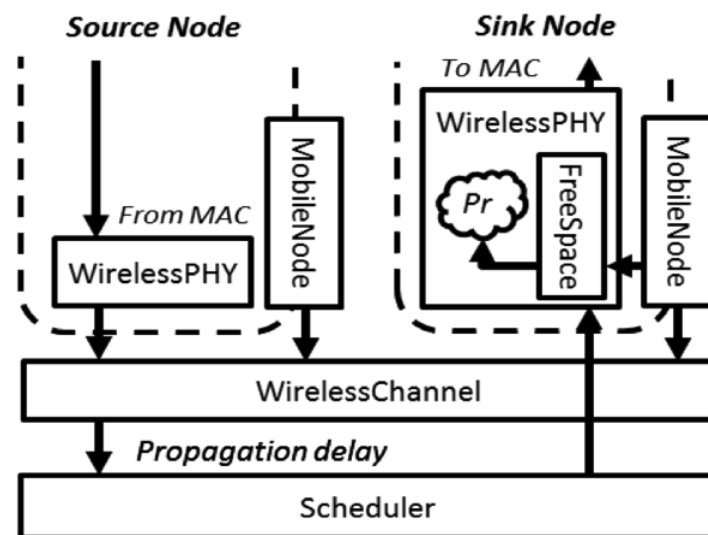


Figure 4-5. The network simulator wireless transmission simulation model (credit: Antoon [6])

Figure 4-5 shows the structure of the ns-2 modules that use the nodes' positions. A node's position has two primary uses in the ns-2 mobile node model. First, the node's position is used to calculate the receive power of the transmissions that a node receives. Each node in ns-2 contains a wireless PHY module that activates when any

node in range transmits a packet. The wireless PHY module contains a propagation module that calculates the packet's receive power.

The propagation module uses the transmitting and receiving nodes' positions to calculate the transmission's propagation distance. For simplicity and due to the direct line-of-sight transmissions in LEO, the FreeSpace propagation module was used, which is based on the Friis transmission Equation 3-1. The FreeSpace module uses the distance between the transmitting and receiving nodes to calculate each packet's receive power.

In order to interface with the external satellites' position databases, a new module, FreeSpaceSTK, was added. Ns-2 modules can define Tcl commands, which Tcl scripts can call to perform module-specific actions. A command handler for the FreeSpaceSTK module was written that logically links an ns-2 node to a corresponding satellite position database. The FreeSpaceSTK receive power calculation uses the satellites' position databases instead of the mobile node positions to calculate node distance. FreeSpaceSTK does not, however, calculate the radio propagation delay.

The second primary use for the ns-2 positioning system is to calculate the radio propagation delay. When the ns-2 WirelessChannel module first receives packets from a transmitting node's PHY, the WirelessChannel module calculates the radio propagation delay for each receiving node by dividing the distance between the nodes in meters by the speed of light (3×10^8 meters per second). To transmit the packet with delay, the WirelessChannel module requests the ns-2's scheduler module to schedule a receive event in the receiving PHY.

Since the propagation methods in the WirelessChannel module are not inheritable, a new module was added, Wireless Channel STK, which duplicated the WirelessChannel module. The WirelessChannelSTK module was modified to use the external satellites' position databases for calculating the distance between the nodes. Using the FreeSpaceSTK module, the WirelessChannelSTK module, and STK-exported constellation information, the modifications allow wireless traffic in ns-2 to resemble traffic between satellites in orbit.

Results

Modifications to the ns-2 version 2.34 installation were applied on Ubuntu Linux 10.10. To run the experimental simulations, two Tcl scripts were created: one script defined the nodes using the flower constellation positions and the other script defined the nodes using the SSRGT constellation positions.

The Tcl scripts specified the protocol for each satellite node's network layers. Ns-2 modules were used for each node's MAC layer since 802.11b-1999 has acceptable long range performance and a wide range of available commercial off-the-shelf hardware. For each node's PHY layer, the standard ns-2 wireless PHY module was used with the propagation configured to use the FreeSpaceSTK module.

The FreeSpaceSTK module was configured in one of the Tcl scripts to use the satellite position database for the flower constellation and the other Tcl script to use the satellite position database for the SSRGT constellation. To simulate the communication channel, the WirelessChannelSTK module was used. The nodes defined by the Tcl scripts behaved like satellites in a constellation and could support traffic from most ns-2 agents, ns-2's representation of a protocol.

A Tcl ns-2 scenario was written to generate sample traffic for the simulation. In the Tcl ns-2 scenario, each non-sink node generated constant bit rate (CBR) traffic over a User Datagram Protocol (UDP) agent to each of the six sink nodes. To prevent the source nodes that were out of range of any sink satellites from transmitting, satellites could detect the presence of sink satellites within 2000 km. Source nodes only transmitted data when they were within 2500 km of a sink node or the ground station in Gainesville, Florida.

Drop-Ratio Verses MAC Slot Times

Since the performance of the MAC layer is significantly affected by the propagation delay of milliseconds between communicating satellites in LEO, simulations were conducted to find optimal 802.11b-1999 MAC module parameters for the scenarios. The standard 802.11g-2007 slot time, the time allocated for a round-trip packet transmission and acknowledgment is 20 μ s, and the short inter-frame spacings (SIFS) is 10 μ s [50]. However, they can be optimized for long-range 802.11. Studies at University of Surrey by [50] extended the slot time to 75 μ s for nodes at 15km apart and later used slot time of 355 μ s for nodes 100km apart [50].

For nodes 2,000 km apart, radio signals propagate one way for a time of over six milliseconds. So, slot times were extended to 7 milliseconds to account for 2,000 km distances. The DCF 802.11g-2007 specification for timing were calculated from Equations 4-2 to 4-4.

$$\text{SlotTime} = \text{AirPropagation} + \text{CCA} + \text{Turnaround} + \text{MAC Processing} \quad (4-2)$$

$$\text{DIFS} = \frac{5}{2} \times \text{SlotTime} \quad (4-3)$$

$$\text{SIFS} = \frac{1}{2} \times \text{SlotTime} \quad (4-4)$$

Since the propagation time increases over these distances, the slot time and SIFS were increased to reduce the drop-ratio. The drop-ratio is the reciprocal of the packet-delivery ratio. For the simulation, SIFS were a function of half of the slot time and the distributed coordination function (DCF) inter-frame Spaces (DIFS) were a function of the varying slot time shown in Equations in 4-3 and 4-4.

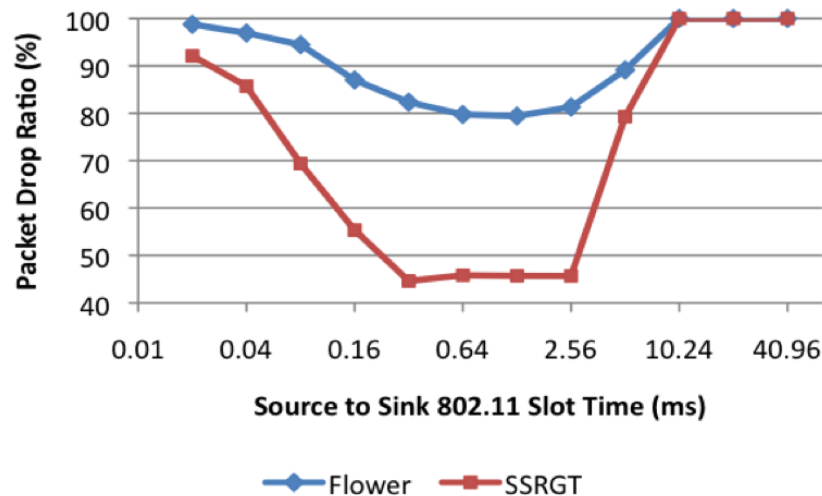


Figure 4-6. Packet drop ratio versus MAC slot time

Figures 4-6 and 4-7 demonstrate that for both flower and SSRGT constellations, the optimal slot time ranges from 500us and 1500us, while the standard 802.11b-1999 slot time causes the MAC to drop nearly all packets.

While the round-trip propagation delay between distant nodes may be higher than 1500 us, the optimum 802.11b-1999 slot time reflects a tradeoff between unnecessary slot time delay for nearby nodes and propagation delay for distant nodes. Since the optimal slot time range is similar for the sink satellite connections to both ground stations and source satellites, sink satellites can use the same physical layer while transmitting to sink nodes and the ground stations.

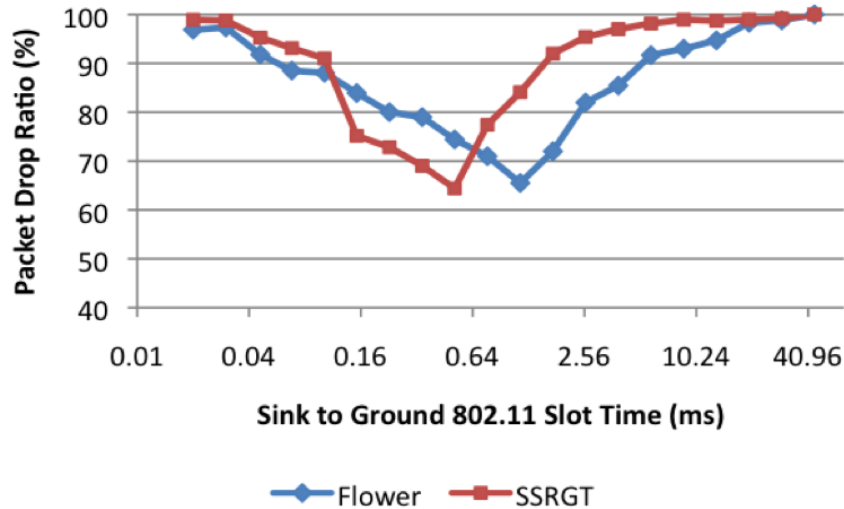


Figure 4-7. Comparison of the ground to sink satellite drop ratio versus the MAC slot time

For the source satellite to sink satellite connections, the SSRGT constellation dropped significantly fewer packets than the flower constellation. With the slot time set at 640us, the SSRGT constellation dropped fewer than 50% of the packets, while the flower constellation dropped more than 75% of the packets. For both the flower and SSRGT constellations, the high drop-ratio in Figures 4-6 and 4-7 demonstrate that any network protocol used by the satellites must perform reliably with intermittent connections.

Throughput Versus Source Traffic Density

The traffic capacities of the flower and SSRGT constellations were tested by simulating the network throughput while increasing the source traffic density (the rate at which the source nodes send packets).

Figure 4-8 shows the network throughput for both the flower and SSRGT constellations between 10 bytes/sec and 80 Bytes/sec, with the source satellites sending 1 Kbyte packets.

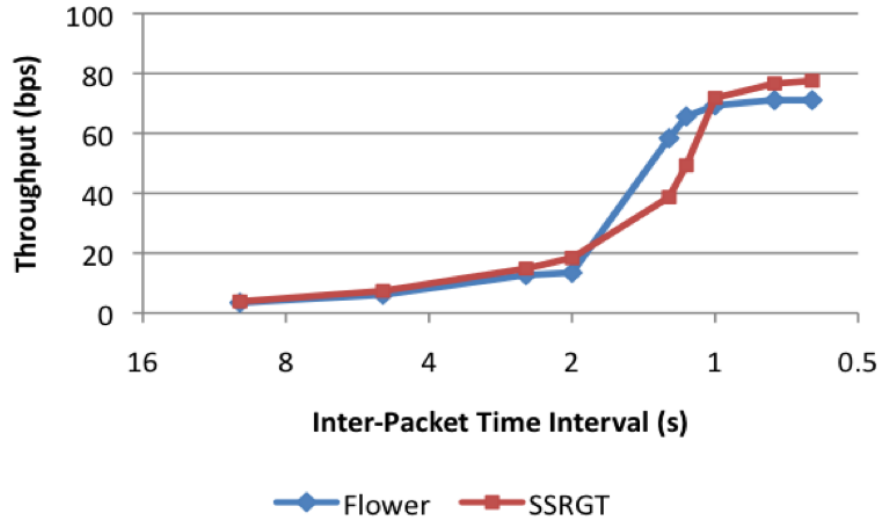


Figure 4-8. Throughput versus source traffic density

Figure 4-9 shows the packet drop-ratio for the SSRGT and flower constellations as a function of source traffic density. The flower constellation maintained a similar throughput as the SSRGT constellation even though the SSRGT constellation dropped fewer packets than the flower constellation.

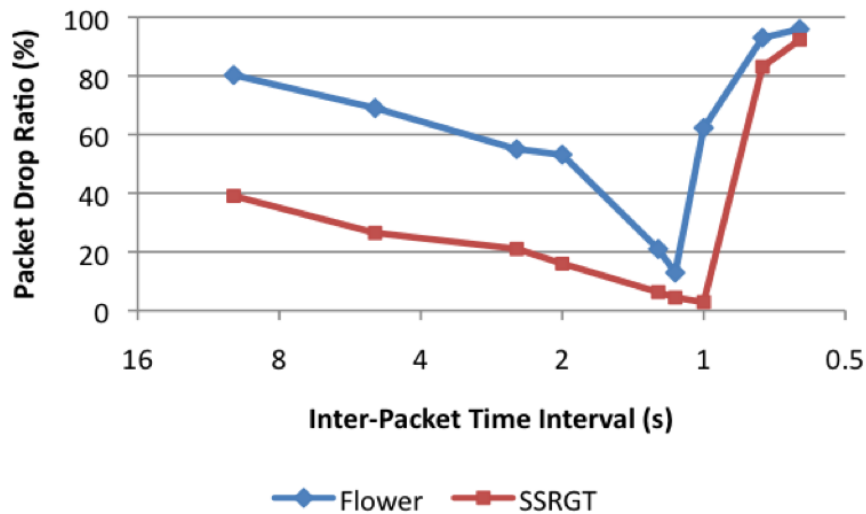


Figure 4-9. Comparison of packet drop ration versus source traffic density

The higher packet drop-ratio for the flower constellation as compared to the SSRGT constellation, combined with the constellations' similar throughputs, suggests

that the flower constellation has more opportunities than the SSRGT constellation to transmit data from long distances. Both constellations suffered very low throughputs, which is expected due to the weak transmitters and long distances involved in the simulation.

Summary

In Chapter 4, a method for comparing the network performance for any constellation was designed in STK based on sink time, drop-ratio, and throughput. Using this method, network performance of two novel LEO satellite constellations was compared: a flower constellation and a sun-synchronous repeating ground track (SSRGT) constellation. In order to compare the constellations' network performance, the Network Simulator (ns-2) used ns-2's mobile node model to simulate complex satellite constellations.

Results revealed that as the satellites opportunistically communicated during a week in simulation time, the satellites in the SSRGT constellation dropped fewer packets than the satellites in the flower constellation. During a period of 500 ms to 1 second, the SSRGT satellite showed a higher throughput.

CHAPTER 5

SIMULATING DTN FOR CUBESATS

Chapter 5 presents a simulation tool to validate DTN protocols for CubeSat cluster topologies and compares the performance of DTN to UDP/IP based satellite constellations. From using this tool, the data rate and access times was determined for CubeSats to form intersatellite links and downlink with ground stations.

DTN Protocol Over CubeSats

The DTN bundle protocol would allow multiple CubeSats to store and forward data for efficient payload data downlinking and ultimately more opportunities to link with a ground station. For longer interplanetary links, DTN's Licklider Transport Protocol (LTP) convergence layer could be used. LTP takes care of high latency, and asynchronous channels because data could be downlinked without many ACKs from destinations. The CubeSats could either be launched into a constellation orbit or cluster orbit. Many DTN implementations have been written for various platforms.

DTN2

For a Bundle Protocol reference implementation, the Delay Tolerant Networking Research Group (DTNRG) created DTN2. DTN2 is described in the Internet Research Task Force standards organization's RFC5050 specification [52]. The codebase from the reference implementation built at Trinity College, Dublin, Ireland evolved into DTN2 [52]. The implementation is hosted on source-forge, open-source, and built primarily in C++. DTN2 includes built-in applications. Users can run a wide range of commands such as dtnping, dtnsend, dtnrecv, and dtnperf.

ION

For a Licklider Transport Protocol (LTP) implementation, the NASA Jet Propulsion Laboratory (JPL) created the Interplanetary Overlay Network (ION) implementation [53]. Specifically, ION implements LTP found in IRTF RFC5325 to 5327 and BP RFC5050. NASA and the University of Ohio built the interplanetary overlay network (ION) as a space oriented implementation. For functionality in space, ION runs as flight software on VxWorks. DTN protocols including BP, LTP, CCSDS File Delivery Protocol (CFDP), and Asynchronous Message Service (AMS) are implemented for ION in the C language [53]. For routing, ION supports contact graph routing (CGR) [53], considered most suitable for routing nodes in space.

IBR-DTN

Built by the Institute of Operating Systems and Computer Networks at Technische Universität Braunschweig, IBR-DTN is a lightweight DTN implementation of the Bundle-Protocol RFC5050. IBR-DTN is particularly useful for an embedded systems involved in intermittent sensor networks [54]. To run efficiently developers compiled using uClibc++, C library optimized for embedded systems. In addition to installing in Linux from the source, IBR-DTN has packages for devices such as raspberry-pi Debian Linux, android-based smart phones, and wireless access points (APs) with OpenWRT.

JDTN

Cisco Systems created a DTN implementation in java known as JDTN. The software is implemented for the mobile android platform. JDTN supports BP RFC5050 and LTP RFC5326.

ByteWalla

The TSLab at the KTH School of ICT in Kista, Stockholm created Bytewalla. The software is written as an adhoc DTN mobile Android application to bring connectivity to remote regions using BP. The application uses mobile phone as data carriers, similar to JDTN.

N4C

Networking for Communication Challenged Communities (N4C) created a DTN2 simulation platform. The platform uses NS-3 to model the physical RF nodal connection. It can also run emulation of DTN at the network layer.

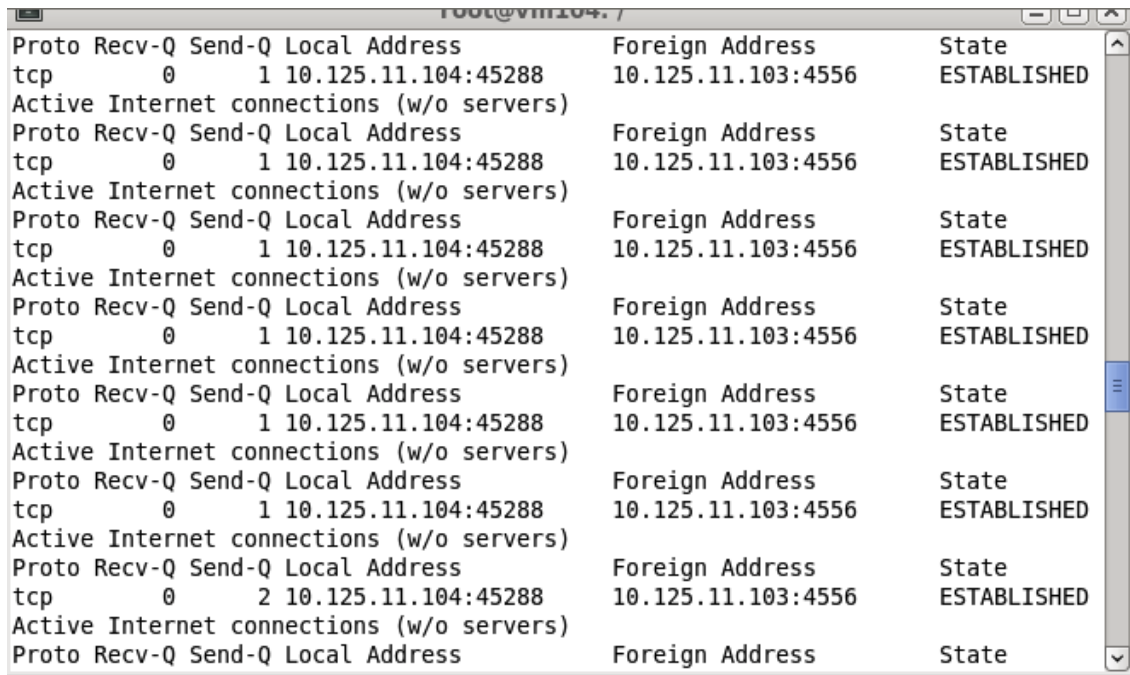
Simulating Delay Tolerant Networking

Simulators such as DTNSim2 [55], and the Opportunistic Network Environment (ONE) [55] have been used in previous studies. DTNSim2 is no longer supported and ONE was developed between 2008 and 2011. Often, these simulation platforms require high computing resources.

The effectiveness of DTN simulators for space inter-networking is sensitive to the ability to achieve a realistic simulation environment. Many DTN simulators do not currently implement realistic channel models. Often, only the network layer is simulated. There is also an issue of cross-simulator comparability. Researchers often create their own simulators to test algorithms, so it can be difficult to compare a new algorithm with existing ones, unless the new protocol is implemented on a variety of simulators. In this work, the Network Simulator 3 (NS-3), a widely available and capable open-source simulator, was chosen for the channel and link layer simulation. Then, on top of the simulated link layer, one of DTN implementations could be run. The goal for the testbed is to experiment with virtual and real hardware nodes.

Design of a Testbed

The following describes the DTN simulation platform, virtual machines, Linux containers, and simulated communication channels. The N4C implementation was picked for the test bed because it had automated network connectivity scheduling. Automated scheduling allowed for simple integration of a DTN network with CubeSat topologies.



Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	1	10.125.11.104:45288	10.125.11.103:4556	ESTABLISHED
Active Internet connections (w/o servers)					
Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	1	10.125.11.104:45288	10.125.11.103:4556	ESTABLISHED
Active Internet connections (w/o servers)					
Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	1	10.125.11.104:45288	10.125.11.103:4556	ESTABLISHED
Active Internet connections (w/o servers)					
Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	1	10.125.11.104:45288	10.125.11.103:4556	ESTABLISHED
Active Internet connections (w/o servers)					
Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	1	10.125.11.104:45288	10.125.11.103:4556	ESTABLISHED
Active Internet connections (w/o servers)					
Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	1	10.125.11.104:45288	10.125.11.103:4556	ESTABLISHED
Active Internet connections (w/o servers)					
Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	2	10.125.11.104:45288	10.125.11.103:4556	ESTABLISHED
Active Internet connections (w/o servers)					
Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State

Figure 5-1. The simulation platform screen capture of the network status (netstat) of the LXC terminal window in real time

The test bed simulated the physical channel as a wireless 802.11g-2007 network with a range of parameters for topology, mobility patterns, data transmission ranges, and data rates. For the higher layers, connections to network nodes that are external to NS-3 were bridged through the host system. Network nodes were in the form of Linux containers (LXCs) and virtual machines. When running the simulation, the network status was viewed in real time for each virtual machine or LXC.

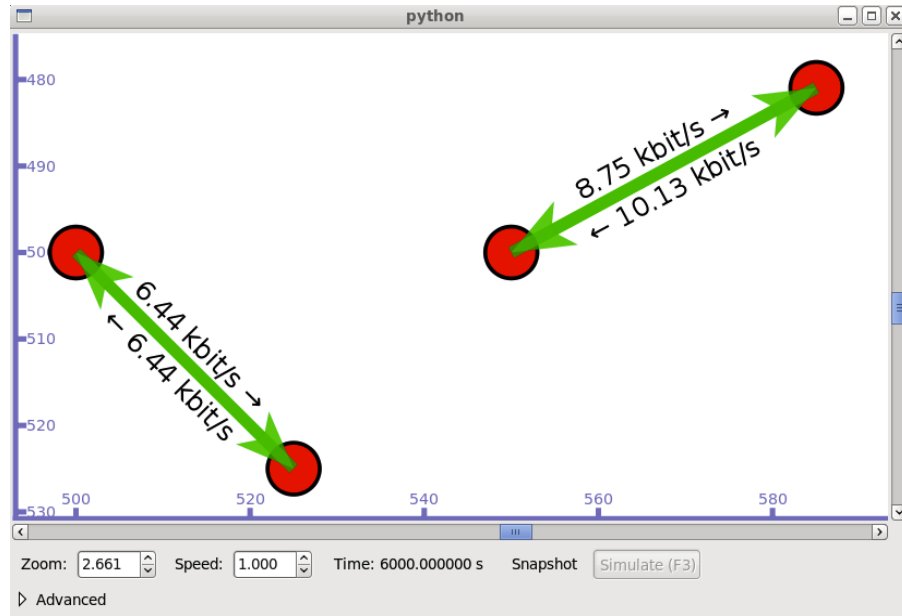


Figure 5-2. A simulation platform screen capture of the python-based visualizer (pyviz) that shows the data rate and node topology. The bottom window shows the network status (netstat) of the LXC terminal window in real time.

The test bed topology, mobility, transmission link directions, and data rate were visualized in real-time and logged using NS-3's python-based pyviz application, shown in Figure 5-2. Note that the visualization shows nodes in two-dimension. However, the node topology can be configured to three-dimensions using Cartesian x, y, z coordinates.

As shown in Figure 5-3, the test bed components include NS-3 and virtual machines with the DTN2 network software stack. NS-3 can model communication channel properties such as delay, transmission rate, error, and packet loss distribution with detailed scheduling. In addition, NS-3 allows configuration of mobility patterns for wireless nodes, networking device properties at the physical and link layer, logging and packet tracing. If required, new models can be constructed and used in simulations.

Simulation Platform

Acting as clients, virtual machines and LXC's connect through NS-3's simulated communication channels. CubeSat flight hardware candidates, in the form of laptops and smart phones with DTN implementations, can connect to the simulated channels through an 802.11 access point created by the host machine.

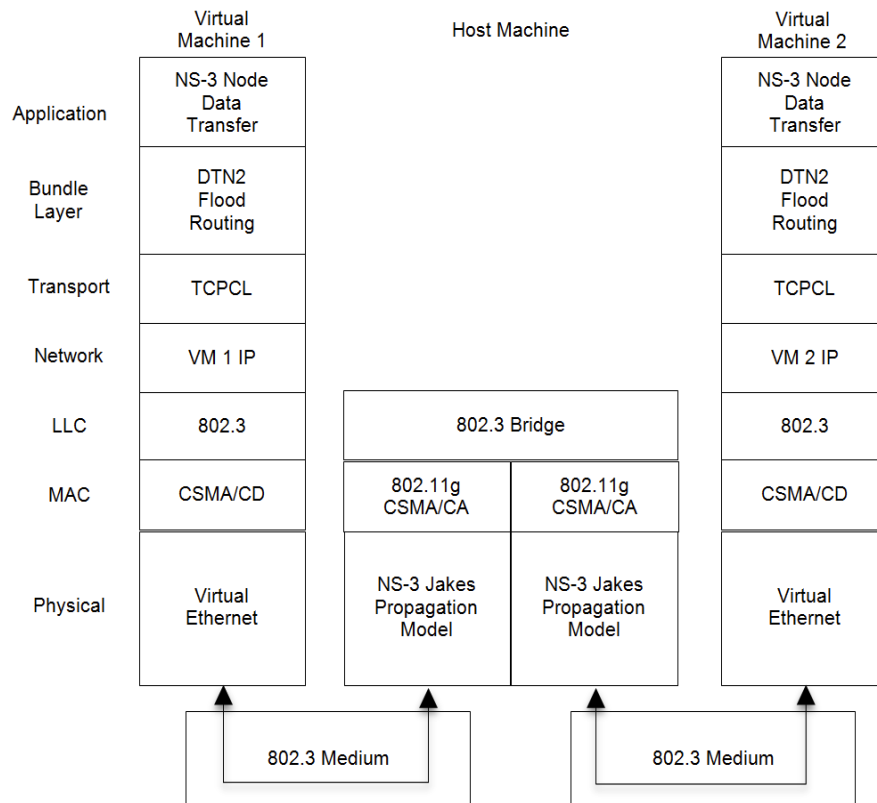


Figure 5-3. Network Stack for transmitting bundles between virtual machines

Virtual Machines

The virtual machines create a native environment for the DTN software. They provide virtual hardware on which operating systems can be installed and configured according to the DTN software requirements with minimal impact to the underlying host. Of the various virtual platforms available, Oracle's VirtualBox was used because, with

available open source editions, modification is feasible. However, VMWare Player and VMWare Server could also be used.

Linux Containers (LXC)

Lightweight virtual nodes have also been implemented in the form of Linux Containers (LXC). Nodes hosted on Linux containers use the operating system kernel of the hosts compared to separate full virtual machines. The resources used for the LXC method is significantly lower than that of virtual machines. Setting up hundreds of LXC as virtual nodes is feasible while still providing the appearance of a guest operating system to the container applications. As a result, a lightweight virtual node, though limited to using the same operating system as the host, requires much less RAM and disk space than a full virtual node and can therefore be replicated in great numbers.

Simulated Communication Channels

The communication channel from ns-3 provides simulated wireless links between the network clients in the physical, MAC, and network layers. In the same way that Linux containers and virtual machines create virtual LAN IP addresses. Remote desktop machines, laptops, and smart phone can tunnel into the simulator. This can be done either by a virtual private network (VPN) tunnel or a wireless access point.

Setup of Experiment

As a main network simulation and server Linux platform, Fedora Linux 64-bit distribution was selected as the operating system both for the host. Ubuntu Linux distribution was chosen due to the DTN2 reference implementation for the guests. Hardware of the host can be any standard personal computer or laptop compatible with the operating system, with at least 8 GB of RAM, and disk space to allow for many virtual machines. Different hardware platforms of interest such as Broadcom or ARM

can be integrated by physically connecting them into the system through an access point. NS-3 comes with a number of channel and device models, including CSMA/CA, WiFi, and WiMax. The DTN2 systems and application software that run on the virtual nodes come from the DTNRG.

The same physical and MAC layers in Chapter 4 were used. The network layer routing and DTN layer was added. Three routing options were available for configuration: static, prophet, or flood. Flood routing was used for all current simulations. The routing configuration is set in the bundle layer as shown in Figure 5-3. To determine the network layer's connections a mobility model from a previous study was used.

A previous study used Java-based orbit propagation and visualization software called SatLauncher to model a CubeSat's mobility upon launch of a cluster [56]. The orbital parameters of each CubeSat were calculated given altitude, radius, rotation rates, mass, and launcher velocity. The velocity in the x-axis and z-axis was calculated.

$$\Delta V_x = \Delta V_S \cos(\varphi) - nR \sin(\varphi) \quad (5-1)$$

$$\Delta V_z = \Delta V_S \sin(\varphi) - nR \cos(\varphi)$$

The angle between the xy-plane, from the positive x-axis is φ . V_s is defined as the vertical kickoff velocity from the launch vehicle surface. R is the radius of the launcher, and n is the rotation rate of the launcher about the y-axis. For the clusters in the test bed, satellites used the 1U CubeSat standard mass and volume, 1kg and 10cm³ respectively. Another goal was creating a cluster of tight formation but maintaining safe distances. The node topologies can be visualized using NS-3's pyviz model or with wireless ranges shown in the network animator in Figure 5-4.

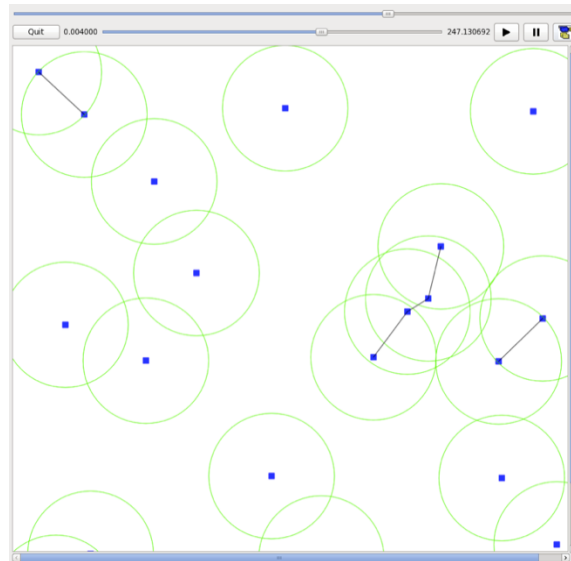


Figure 5-4. Network Animator (NetAnim). Circular rings show a node's maximum transmission range

Results

Simulations were run 10 times, generating bundle at four nodes and derived data rates for two distances 50m and 2,000 km. Note that the default DTN configuration in the virtual machine uses a UDP convergence mechanism, but to the simulation platform the traffic generated is similar to IP traffic. Figure 5-5 shows the topology for three sensing nodes transmitting to a sink node.

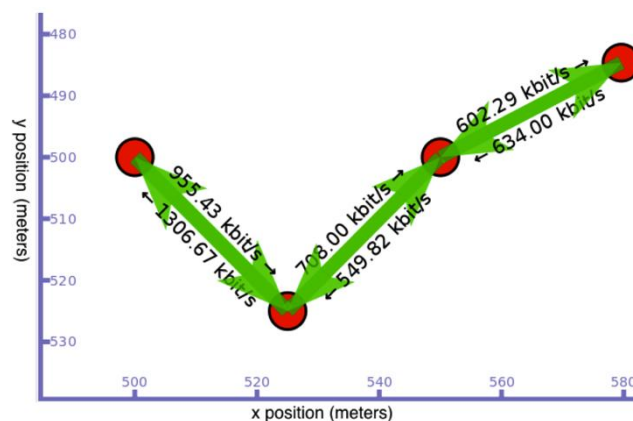


Figure 5-5. A python visualization graphical user interface of an 802.11g-2007 topology, DTN-enabled nodes for a starting spacing of 50 meters, average data rate for this topology is 755.84 Kbit/sec.

The sensing nodes transmit from 802.11g channels at a spacing of 50 meters. The resulting average data rate for this topology is 755.84 Kbit/sec. For a cluster of Wi-Fi nodes with spacing of 2,000 km, the average resulting data rate is 10.44 Kbit/sec. Table 5-2 shows the data rate increased as compared to the experiments in Chapter 4. The results show the platform is capable of modeling various connectivity patterns, and that DTN protocols can effectively help data flow across a fractionated network.

Table 5-1. Comparison of the average data rate between DTN and UDP clusters

Environment	Average Data Rate
DTN 50 meters	755.84 Kbit/sec
DTN 2000 kilometers	10.44 Kbit/sec
DTN 2000 kilometers	80 bit/sec

Summary

Chapter 5 describes a new virtual space inter-networking environment and testbed. The testbed investigated networking performance of satellite constellations and clusters using DTN protocols. The first topologies for simulation virtual challenged space communication environments are described. Then, a testbed of nodes with Linux-based hardware was used to emulate CubeSats. For experimentation, CubeSat cluster mobility models, and topologies were evaluated using the DTN metric of data-rate. When compared, the DTN data rate outperformed UDP/IP with higher data-rate for the given mobility models.

CHAPTER 6

PERFORMANCE COMPARISON OF DTN PROTOCOLS FOR HIGH DELAY SPACE BASED OPTICAL CHANNELS

As missions in space expand from space-to-ground station links to multiple relay spacecraft, orbiting the Moon or Mars, the Consultative Committee for Space Data Systems (CCSDS) [57] has begun to standardize Delay Tolerant Networking (DTN) protocols to support store and forward space communication missions. Recently, CCSDS is standardizing the Bundle Protocol (BP) and the Licklider Transport Protocol (LTP) [58]. These DTN protocols address issues that may occur in store and forward topologies including high latency, and lack of end-to-end connectivity.

DTN protocols have been used for past space demonstration missions, including the Deep Impact Network Experiment (DINET) on the EPOXI spacecraft (2008), the Cisco router in Low Earth Orbit (CLEO) on the UK-DMC (2008), the International Space Station (2009), and the internet router in space (IRIS) on IntelSat-14 (2011) [59]. Advanced DTN concepts, such as reactive bundle fragmentation, which breaks data into independent fragments when encountering disruptions, proved beneficial as well. However, these past missions are limited to data rates under tens of megabits per second because of the transmitter's channel bandwidth and power constraints.

Several current missions, such as NASA's Lunar Laser Communication Demonstration (LLCD) and planned missions such as NASA's Laser Communication Relay Demonstration (LCRD, 2017) will rely on optical communication, which raises the requirements on DTN networking protocols. When compared to radio frequency (RF) communications, optical systems allow for higher bandwidth, power-efficient links. For example, the LLCD mission downlinks at 622 Mbit/second with a range of approximately 400 thousand kilometers [60]. Optical links have the same speed of light delays but are

more vulnerable to disruptions when compared to RF. Applying DTN protocols to optical communications eases the potential problem of intermittent connections. LCRD adds DTN's bundle protocol to LLCD's hardware.

Table 6-1. DTN Space Mission Demonstrations

Mission Demonstration Name	Date
Epoxi – Deep Impact Network Experiment	2008
UK-DMC – Cisco Router in LEO (CLEO)	2008
IntelSat-14 – Internet Router in Space (IRIS)	2009
International Space Station – Commercial Generic Bioprocessing Apparatus (CGBA)	2009
Multi-Purpose End-To-End Robotic Operation Network (METERON)	2012
Edison Demonstration of Smallsat Networks	2014
Laser Communication and Relay Demonstration (LCRD) - Optical	2017
Integrated RF and Optical Communications (iROC) - Optical	2017

While previous DTN-based testbeds experimented with bandwidth capacity in megabits, Chapter 6 presents a channel emulation testbed that supports multiple gigabit-bandwidth Ethernet clients and servers simultaneously. A testbed with high bandwidth capacity was designed to support multiple client machines using gigabit Ethernet links. When placed in-between client machines, the testbed emulated optical channels with high propagation delay, and bit errors for free-space optical uplinks and downlinks that vary as a function of time. Client transmitted using bundle protocol, Licklider Transport Protocol, and TCP. Goodput was observed to compare the different protocol configurations over several iterations of round trip times by increasing the propagation delay. Additionally, for flexibility virtual LANs and link aggregation was used, which allowed configuration for various network topologies.

First, Chapter 6 details previous DTN protocol performance studies. Then, a theoretical analysis of TCP throughput for increasing round trip times, the testbed

design and experiments, results, summary, and lessons learned from the testbed are presented.

DTN Implementation Performance Studies

For an overlay store and forward network, RFC 5050 specified the bundle protocol as a layer that lies between the application and transport layers. The BP layer encapsulates bundles, a series of contiguous data blocks, into an underlying convergence layer (CL) adapter, shown in Figure 6-1.

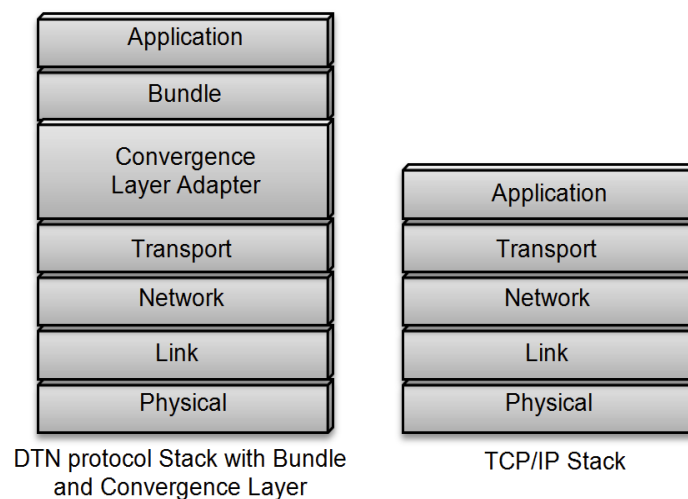


Figure 6-1. Testbed system design with convergence layers

Common convergence layer adapters for DTNs include a Transmission Control Protocol CL (TCPCL) [61], User Datagram Protocol CL (UDPCL) [61], and Licklider Transmission Protocol (LTP) [62]. TCPCL allows for reliable transfer of frames through a TCP/IP network [52], and UDPCL is the unreliable transport through user datagram protocol [61]. Applied to the space links, LTP allows for long delay, point-to-point channels [63].

Various testbeds have emulated space communication channels to measure DTN protocol performance. There are advantages and disadvantages of testbeds to

characterize DTN, including the Space Communication and Networking Testbed (SCNT) [64], the Space Internetworking Center (SPICE) [65] DTNRG's DTNBone [66], and the Jet Propulsion Lab's optical testbed [67].

Wang at Lamar University designed SCNT as a PC-based testbed to characterize DTN protocol goodput with latencies up to five seconds [64]. The SCNT refers to the entire testbed including clients equipped with the ION implementation of DTN. A single centralized machine called space link simulator (SLS) emulates the wireless channels for the entire network through virtual instrumentation in Labview software. The SLS emulates channel delay variations, asymmetric link ratios, and bit error rates (BERs) by additive white Gaussian noise generation. The SCNT has successfully tested many DTN protocols over cislunar delay, and asymmetric link simulations. However, the maximum baud rate for the testbed is 115,200 bits per second [64]. Inducing any delay or asymmetric links lowers the maximum data rate further according to the bandwidth delay product. While, the SCNT's bandwidth capacity met the requirement of current satellite system simulations, the testbed would need further development to simulate future satellite communications.

The Space Internetworking Center (SPICE) in Thrace, Greece built a DTN testbed of 12 nodes in three different lab locations: Democritus University in Thrace, Hellenic Aerospace Industry in Athens, and Massachusetts Institute of Technology in Cambridge, Massachusetts [65]. SPICE experiments with CFDP, AMS, and space packet protocol by transmitting files between these labs. SPICE also communicates with HellasSat, a geosynchronous telecommunications satellite, as a relay node. As opposed to SCNT's centralized space link simulator, SPICE distributes channel

emulation throughout the testbed. Each of the 12 nodes that connect to the testbed utilizes network emulation functionality (netem), included in Linux kernels. The command line tool, traffic control (tc), in the package IProute2 tools, configures network emulation. With these tools, each node in the testbed can emulate bandwidth, packet error rate, corruption, duplication, re-ordering, and delay. Distributing emulation across the testbed eliminates processing overhead that could occur on a centralized channel-emulating machine when the number of network nodes increases. This allows for a highly scalable testbed. However, netem and tc are only available on particular Linux distributions. Thus, the testbed lacks compatibility for client devices other operating systems such as RTEMS and VxWorks, real-time systems used on most spacecraft. In addition, SPICE designed a graphical user interface (GUI) to track and control the link emulation parameters for each of the nodes. However, the GUI only supports the ION implementation currently.

DTNbone defines itself as a collection of nodes worldwide running DTN bundle agents and applications [66]. DTNbone users can test connections using Ohio University's and Glenn Research Center's Always On networks of ION and DTN2 nodes, which have a defined schedule of disconnections in the topology, described in [66]. However, Beuran characterizes DTNbone as an interoperability testbed, given that it contains five different DTN implementations [68], as opposed to a DTN performance testbed. For performance testing, the testbed would most likely need to be local.

Schoolcraft first characterized DTN protocols for optical communication in a testbed of two PCs in [67]. The PCs used ION with LTP's Datagram Retransmission (DGR) [67]. Similar to TCP, DGR has an adaptive timeout interval congestion control,

but without the round trip frequency of TCP. The testbed created a unidirectional forward optical link using two Perle media converter systems to convert gigabit Ethernet links into fiber and free space mediums. For free-space optical links, a variable rotating attenuator could reduce the signal from 0 to 10 dB, which simulated the intermittent connections experienced. To create an asymmetric link, the testbed used Ethernet with rate limited at the Internet protocol (IP) by modifying Linux kernel networking queueing settings. The testbed successfully emulated an optical link with channel disruptions, and asymmetric bandwidth. However, the testbed did not examine long propagation delays to validate DTN performance over high latency channels.

In [69], Pottner compared performance of the three common DTN bundle protocol implementations between two PCs. The experiment compared implementation throughput for memory-based versus persistent disk-based storage, and varied bundle sizes. When using DTN2 as the same transmitting and receiving implementation, the highest throughput was 687.329 Mbit/second with 1 MB payload size. However, the experiment did not emulate the channel latency, rate, or BER.

Theoretical Analysis

To model the upper bound of the bundle protocol with the TCP convergence layer throughput, the theoretical limitations of TCP throughput were calculated. Equation 6-1 shows the Mathis Equation [70], which sets the upper bound for TCP's theoretical throughput.

$$\text{Maximum Throughput} \leq \frac{\text{MSS}}{\text{RTT}} \times \frac{C}{P} \quad (6-1)$$

Equation 6-1 leverages the maximum segment size (MSS), round trip time (RTT), packet loss (P), and a constant that incorporates a random or periodic loss model and

an acknowledgement strategy (C). The maximum frame segment size derives from the Maximum Transmission Unit (MTU). For Internet applications, the common MTU is 1500 bytes (to calculate MSS, the TCP overhead of 40 bytes for IP and TCP header data is subtracted from the 1500 byte MTU). Assuming an average random loss of 10^{-6} for P, with delayed acknowledgments, the value of the Mathis constant, C, is 0.93.

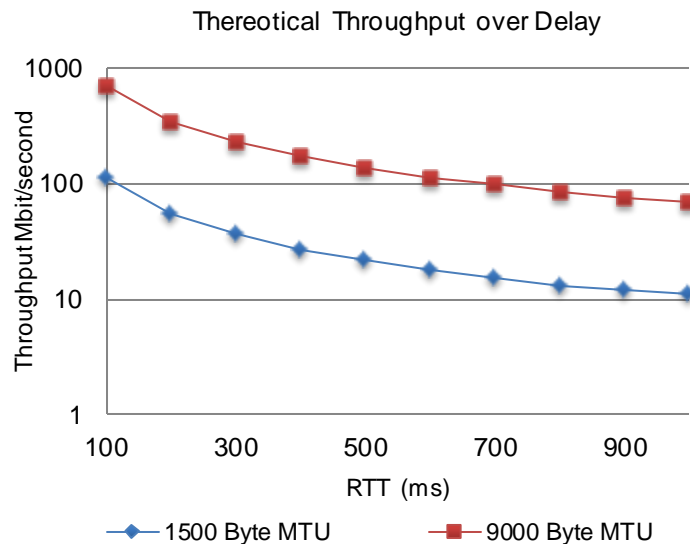


Figure 6-2. Theoretical Throughput over Delay

Figure 6-2 shows the theoretical bandwidth for TCP with BER of 10^{-6} over various round trip times. To allow for higher rates along mediums with increasing RTTs, network interfaces were configured for jumbo frames with an MTU of 9000 bytes. Increasing the frame size reduces interrupts processed by the CPU. In addition, the required overhead, header to data ratio, decreases from 2.6% to 0.44%. With a typical geosynchronous round trip time of 500 ms, the maximum throughput of a 1460 byte, and an 8960 byte MSS calculates to 21.72 Mbit/second, and 133.325 Mbit/second respectively. Thus, the expected upper limit of the bundle protocol using the TCP convergence layer is the theoretical maximum TCP throughput with a 9000 byte MTU.

Testbed Design

After surveying previous DTN protocol testbeds, certain characteristics were chosen from each study. A centralized emulator similar to SCNT [64] was designed to allow simple system configurations and control, but added an Ethernet switch for a more flexible and scalable system, shown in Figure 6-3. The testbed's centralized channel emulator allows for any type of client device operating system to connect. The testbed used the network emulation functionality in Linux similar to SPICE [65] to natively emulate wide area network properties such as delay, and loss, but parameters could vary as a function of time. Last, a testbed for demonstrating BP's maximum throughput was developed, compatible with three common DTN implementations (ION, DTN2, and IBR-DTN). This is similar to Pottner's study [69], but added wireless channel emulation features such as variable delay, and modeled BER over an optical channel as a function of time.

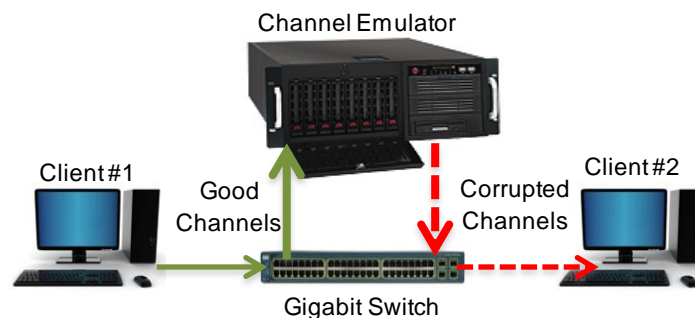


Figure 6-3. Testbed System Design

Architecture

The testbed's centralized network emulation software was on a Concurrent ImaGen server with a 2.8 GHz quadcore Xeon 5600 running CentOS 6.2. Theoretical maximum throughput, as shown in Figure 6-2 with 9000 byte MTUs and less than 500 ms RTT, requires gigabit bandwidth capacity for each channel link. Therefore, another

requirement for the system was to support multiple clients and servers simultaneously. To allow gigabit links and multiple devices, two four-port Ethernet gigabit network interface cards were installed in the ImaGen server. The eight Ethernet ports on the Imagen server were connected to a 24 port Cisco Catalyst 3560G switch. The ports on the Imagen server were channel bonded, as shown in Figure 6-4.

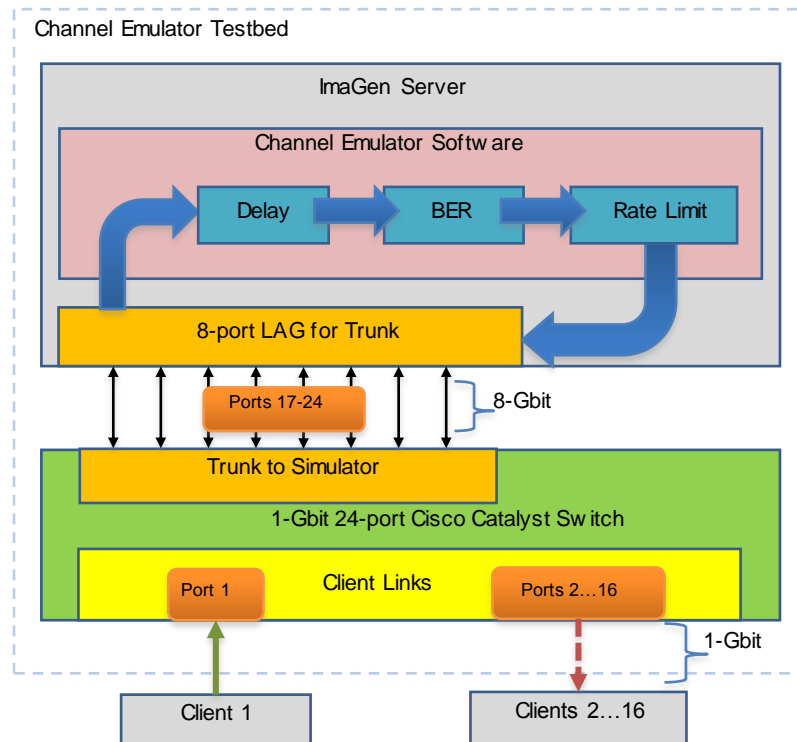


Figure 6-4. Testbed Architecture

Channel bonding, also known as link aggregation (LAG), is a technique in which more network interfaces combine on a host computer for redundancy or increased throughput. Linux kernels now come equipped with a channel bonding driver. There are different modes of channel bonding. For increased throughput, a balanced round-robin policy was chosen. In balanced round robin, packets are transmitted in sequential order from the first available slave interface to the last. Balanced round robin also provides load balancing and fault tolerance. All ports on the host machine are set to 9000 byte

MTUs and promiscuous mode, which mandates that all traffic the port receives, shall pass to the central processing unit as opposed to only the frames the port anticipated to receive.

The Cisco switch calls channel bonding Link Aggregation Control Protocol (LACP). Once ports 1-8 were configured for the Cisco switch to LACP, the testbed had eight gigabits of total bandwidth capacity. For Clients, Dell PowerEdge 2950s with quad core Xeon 5300 2.8GHz CPUs were used. These clients connected to the testbed through the ports 17-24 on the Cisco switch. The switch mapped the 16 ports as virtual LANs (VLANs). With VLANs, devices behave as if they connect through a single, network segment. This allows for a flexible and fast testbed with up to 16 users, and eight-gigabit bandwidth capacity.

Channel Emulation Software Parameters

There are three basic channels for communication between a flight terminal and ground station. In the flight terminal's payload channel, BER is negligible, but delay and jitter change over time depending on the flight hardware. When the signal moves through the wireless space channel, BER changes over time with atmospheric changes of either RF or optical signals. For example, a geosynchronous (GEO) orbit fixes delay to approximately 500 ms with perturbations. Jitter for the space channel would be negligible. Similar to the payload channel, the ground station channel has negligible BER, but delay and jitter have changes over time.

For testbed purposes of simulating a ground station to flight terminal channel, bit error rate, rate limit, delay, and jitter were emulated as functions of time. Jitter is defined as a change in delay brought upon from hardware processing at the transmitting and

receiving terminals. Table 6-2 summarizes the effects of a ground station channel, space channel, and payload channel.

Table 6-2. Ground Station Channel, Space Channel, and Payload Channel Parameters

Parameter	Ground Station Channel	Space Channel	Payload Channel
Bit Error Rate	Negligible	Atmospheric conditions for RF and optical	Negligible
Delay	Small change with processing	500ms for GEO, 20ms for LEO, small change with Doppler	Small change with processing
Jitter	Small change with processing	Negligible	Small change with processing

The Channel Emulator operated as a link layer bridge (not as an IP router). Intercepting data at the link layer allows for no special IP addressing in the network layer. The source and destination devices will behave as if they are directly connected. Linux network emulator provides basic emulation. Command scripts changed the emulation parameters over time to a resolution of one millisecond.

Modeling Optical Atmospheric Conditions

There are several dynamic effects on free-space optical channels through the atmosphere. Atmospheric scintillation is the first and fastest effect. Varying temperatures through the atmosphere and index of refraction causes the light to focus and defocus at random, time varying ways. In this case, the effective power received can vary over a range of 20 dB and the time scale of the variation is tens of milliseconds [71]. This effect never really gets up into the one-second range. Employing coding and interleaving, helps mitigate this effect.

Another dynamic element that comes into play is the pointing and tracking systems on each of the terminals. For an optical link, the narrow beams require high fidelity pointing of both telescopes. Even small perturbations in a payload such as vibrations due to the reaction wheels, and maneuvering the solar panels can result in mispointing, which causes the received optical power to dip. The time scale here is similar to the scintillation, though in worst-case analyses there may be some resonances that allow some mispointing effects to last for a second period. Coding and interleaving helps here, but the longer-term resonances may overwhelm the interleaver's ability to whiten the effect. In addition, the spatial tracking loop can fail, in which case the telescope's control electronics perform re-acquisition. The period for this case may take tens of seconds [71]. If the clock synchronization fails, this could require seconds to recover, which is not fast enough for coding and interleaving to handle the majority of it.

Longer-term outages will likely occur due to cloud blockages, for example. These likely occur on the time scale of minutes. Coding and interleaving is completely ineffective in these cases. This is where the networking protocols, handover to alternate ground stations, and store and forward techniques have a significant impact.

Thus, for the testbed, models for cloud blockages were created. Longer-term outages for uplinks and downlinks were modeled. To start a 2-state Markov model was used with a good case of 10^{-9} BER and bad case of 10^{-6} . Later, BER was modeled from commercial optical hardware parameters in RSoft OptSim with 1 ms resolution.

Turbulence was moderately strong, with a measured optical turbulence (C^2_n profile of

2x clear 1) [71]. BER ranged from 10^{-9} to 10^{-2} over this time. The BER modeling included FEC and interleaving corrections.

Channel Emulator Experiment Setup

The ION implementation was installed on a client machine, and used the bpsendfile executable to send through the channel emulator testbed to another machine as pictured in Figure 6-5. BP/TCPCL over TCP/IP sent files of 10MB, 100MB, and 1GB transferring bundles of sizes 1MB, 10MB, and 100MB respectively, so each file transmission sent ten bundles. The channel emulator intercepted, queued, and corrupted bits according to the BER profile, then passed data through to the destination. The channel emulator created a relayed optical communication scenario. A ground station uplinked data to a flight terminal, and the flight terminal downlinked data back to the ground station.



Figure 6-5. Setup for the channel emulator experiment at Goddard Space Flight Center. Photo is courtesy of Paul Muri.

Round trip time was increased to a maximum of one second to compare varying bundle sizes. Goodput, the destination node's application layer data, was measured using Wireshark, a common network protocol analyzer. As a benchmark, standard

TCP/IP packets were sent through the channel emulator using iperf, a tool for measuring maximum TCP and UDP bandwidth performance. The DTN implementation was compared to TCP/IP alone. In another experiment, BP/TCP over TCP/IP was compared to BP/LTP over UDP/IP, with 1 MB bundle sizes, to TCP/IP. In this comparison, RTT was increased to a maximum of 16 seconds.

ARM Processor Test Setup

To test the DTN ION implementation on hardware that can feasibly run on a CubeSat, the DTN ION implementation was cross-compiled for ARM-based development. The bundle protocol on ION was ported to an AT91SAM9G development board shown in Figure 6-6. The AT91SAM9G board was specifically chosen because it has an ARM-based CPU, complies with the CubeSat volume, mass, and power limitations, and comparable to new commercial CubeSat command and data handling boards [72].



Figure 6-6. The AT91SAM9G board with serial cable interface. Photo is courtesy of Paul Muri.

The bundle sizes varied from 100 bytes to 65 Kbytes and measure system usage, throughput, and transmission unit utilization. For CPU utilization, the iostat

command ran on the AT91 processor during bundle transmissions. `lstat` monitors “system input/output device loading by observing the time the devices are active in relation to their average transfer rates” [73]. `lstat` measured averaged user level (application), system level (kernel), and total CPU utilization in one-second intervals, during the execution of the ION implementation with bundle protocol and TCP convergence layer. Throughput was measured for bundle sizes ranged from 100 bytes to 65 Kbytes. The board connected to the channel emulator and round trip time delays increased from 10 ms to 1 second on bundles sizes of 65, 30, and 10 Kbytes. Kbytes per transmission unit was also measured when the MTU is set to jumbo frames [74], or 9000 bytes.

Results

Network Goodput for Varying Bundle Size

Results plotted in Figure 6-7 showed that BP/TCPCL/TCP with 100 MB bundle size goodput surpassed TCP/IP at 25 ms delay with a goodput of 445.3 Mbit/second compared to TCP’s 432 Mbit/second. At 50 ms delay, TCP/IP goodput fell to 193 Mbit/second and BP with bundle size of 10 MB transmitted at 340.9 Mbit/second goodput. Then at 200 ms, 1 MB bundle sizes surpassed TCP/IP with a goodput of 61.1 Mbit/second compared to TCP’s 47.7 Mbit/second. At a full second RTT, TCP dropped to 1.46 Mbit/second, while the 1, 10, and 100 MB bundle sizes transferred at 18.8, 27.4, and 37.1 Mbit/second respectively. The measured goodput stayed within 10 Mbit/second between bundle sizes of 1 MB and 10 MB after 300 ms RTT, showing little difference in goodput for long RTTs. 100 MB bundles had more significant goodput increases over longer RTTs, and came the closest to the maximum calculated theoretical bandwidth.

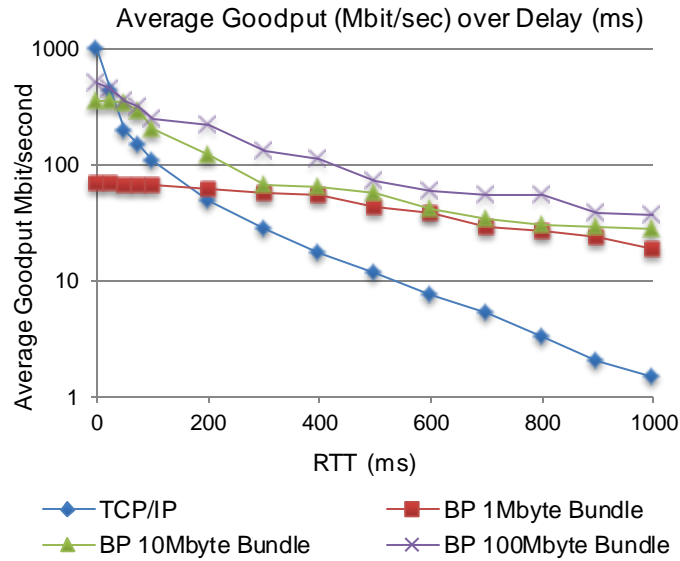


Figure 6-7. BP/TCPCL with varying bundle sizes vs. TCP/IP

BP/TCP for High Latency

Performance was tested for BP/LTP/UDP, BP/TCPCL/TCP, and TCP/IP with increasing round trip time up to 16 seconds. TCP and BP/TCPCL dropped below one megabit goodput at 1.25 and 5 seconds delay respectively.

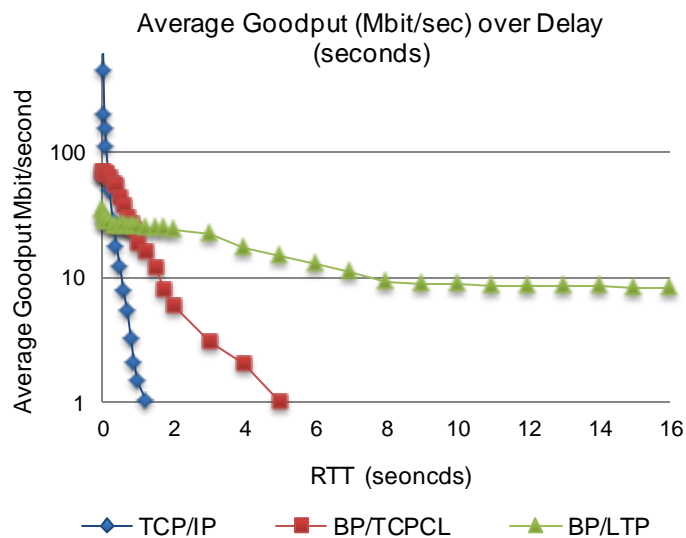


Figure 6-8. TCP/IP, BP/TCPCL, and BP/LTP goodput over added delay ranging from 0 to 16 seconds

BP/LTP surpassed BP/TCPCL in goodput when the delay was set to one second and longer. BP/LTP stayed above eight Mbit/second mean goodput up to the maximum tested delay.

AT91SAM9G Resource Usage

After experimenting with the channel emulator on Intel Xeon processors, an ARM microcontroller was benchmarked as a viable hardware option for CubeSat DTN hardware. The AT91SAM9G board is considered a candidate for CubeSat command and data handling, so the DTN ION implementation was ported to the development board. The optimum bundle size for maximizing CPU usage was found using iostat while running ION bundle protocol with a TCP convergence layer. The throughput change was measured with and without added delay from the channel emulator, and frame utilization efficiency was measured while varying bundle sizes.

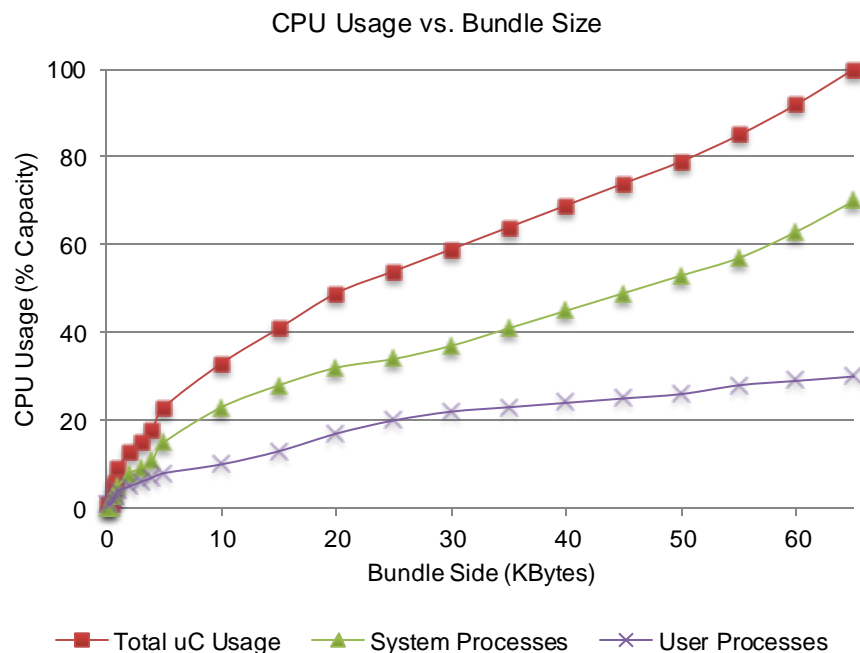


Figure 6-9. Microcontroller used resources used by ION for a 65 Kbyte bundle size using BP/TCP

Figure 6-9 shows the resources used by system, user, and total processes for varying bundle sizes. Since the AT91 processor was 100% utilized running bundle protocol with the TCP convergence layer on ION when transmitting at 65 Kbytes bundle sizes, 65 Kbytes the maximum bundle size was used. Utilization was then observed by stepping down bundle sizes size from 65 Kbytes. At 65 Kbyte bundles, system to user utilization was 70% to 30%. At 30 Kbytes sized bundles, total utilization was 59% with 37% system and 22% user. At 10 Kbytes sized bundles, total utilization was 33% with 23% system and 10% user. Thus, the system utilization increased at a slightly higher rate than user when increasing bundle sizes.

Throughput per Bundle Size

Figure 6-10 shows performance of throughput increased linearly when increasing bundle size from 100 bytes to 65 Kbytes. 65 Kbyte bundles yielded a maximum throughput of 25.64 Mbit. The larger the bundle, the more efficiently software can transmit. The efficiency is even more apparent when delay is increased between the source a destination nodes.

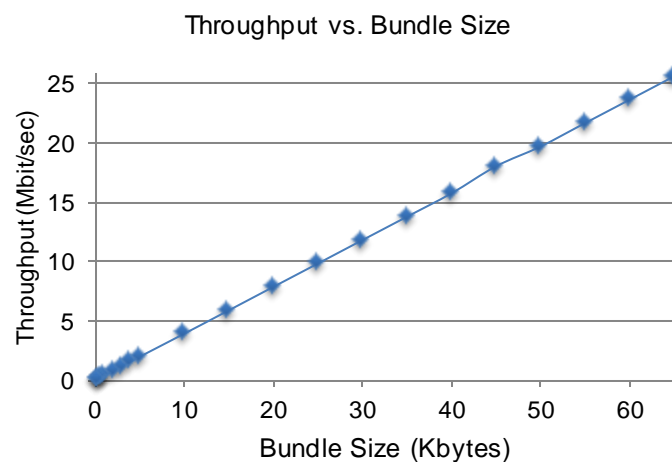


Figure 6-10. AT91SAM9G throughput for bundle sizes of 100 byte to 65 Kbyte using BP/TCP

Figure 6-11 shows how running BP/TCP on the AT91SAM9G is resilient to propagation delays. The bundles with sizes of 65, 30, and 10 Kbytes transmitted through the channel emulator with added round trip time ranging from 10 ms to 1 second. For a typical low earth orbit round trip time of 50 ms [75], bundle sizes of 65, 30, and 10 Kbytes yielded throughput of 24.84, 11.71, and 3.86 Mbit/sec respectively.

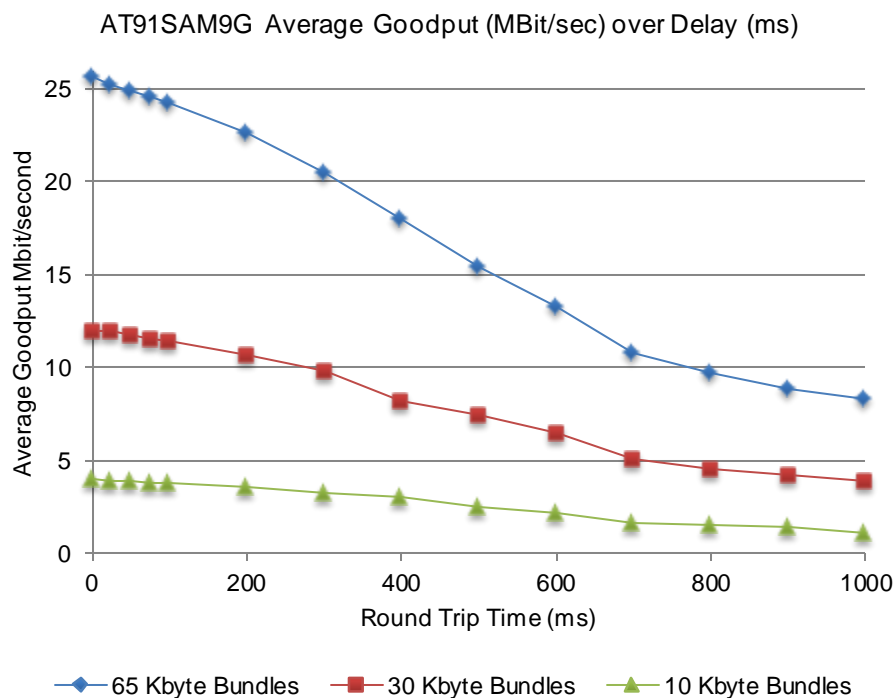


Figure 6-11. Average throughput for 65Kbyte, 30Kbyte, and 10Kbyte bundles with RTT increasing from 10ms to 1 second

Data Utilization per Transmission Unit

The AT91's MTU were set to jumbo, 9000 byte, frames. Figure 6-12 shows frame utilization ramps up quickly as bundle size increases from 100 byte to 10 Kbyte bundles. At 10 Kbytes bundle sizes, the transmission unit data averages 6,345 bytes. The frame utilization then flattens out by 30 Kbyte bundle sizes when the frames average 8,546 bytes of data per transmission unit. As a result, frames are under utilized until the bundle size reaches 30 Kbytes.

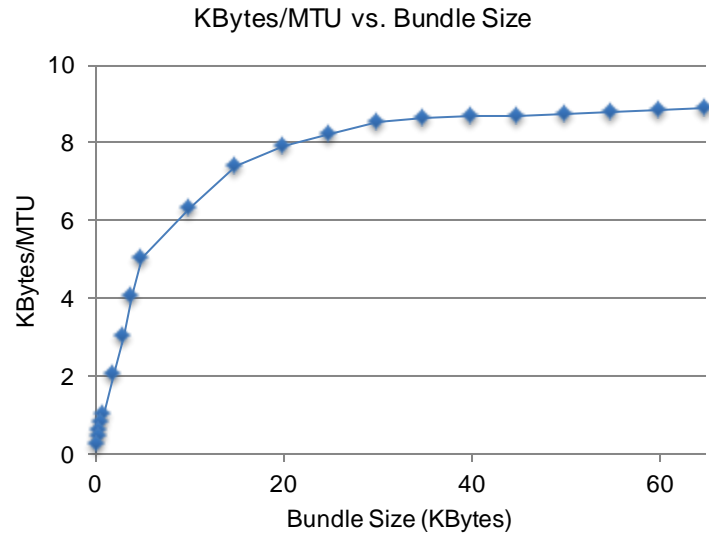


Figure 6-12. Data (in Kbytes) per each transmission frame with a 9 Kbyte MTU using BP/TCP

Summary

In conclusion, a testbed that could measure the performance of DTN protocols through optical, gigabit bandwidth channels was designed. The testbed contained a centralized channel emulator that created optical and RF propagation delay, asynchronous channel rates, and bit errors over the ground station channel, space channel, and payload channel. The channel emulator modeled delay, and bit errors as a function of time as separate channels for uplinks and downlinks. The maximum goodput of various DTN protocols and bundle sizes was measured.

As a result, the testbed demonstrated that using the bundle protocol on top of TCP surpassed the bandwidth when RTT increased beyond 200 ms. Large bundle sizes of 10 MB and 100 MB transmitted most efficiently and came closest to the theoretical TCP bandwidth. The theoretical calculation from the Mathis Equation did not account for when retransmission timeouts occurred. This is one of the main reasons the result diverged from theoretical. Every DTN implementation, DTN2, ION, and IBR-DTN would

use most of the CPU resources when transmitting bundles of 65 kilobits. The channel emulator machine was also CPU constrained when creating long delays at high rates.

The AT91SAM9G was also contained to bundle sizes of less than 65 KBytes, and the maximum throughput for transmitting these bundles was 25.64 Mbit/sec. When a typical low earth orbit round trip time of 50 ms was added, the throughput decreased to just 24.84 Mbit/sec. Thus, the bundle protocol was able to overcome round trip delays of 50ms, with a low decrease to throughput, even on a low-powered ARM microcontroller.

CHAPTER 7 CONCLUSIONS AND FUTURE WORK

This investigation and implementation of a CubeSat communication networking protocol included taxonomy of intersatellite linking distributed systems, CubeSats, and antennas presented in Chapter 2. Chapter 3 presented experiments and simulations for CubeSat antenna candidates. LEO satellite orbits were analyzed as networking topologies in Chapter 4. In Chapter 5, DTN protocols were incorporated into simulations to evaluate the performance for an RF 802.11 channel. Then in Chapter 6, a testbed was created for an optical channel medium emulation. In addition, DTN protocols were implemented into a viable CubeSat Arm-based development board.

Major contributions of this dissertation include a trade study which characterized large-scale satellite clusters and constellations launched or designed. Then design specifications were described for CubeSats launches. From the studies, four main issues of internetworking CubeSat for space were noted: RF allocation, Orbital Properties, Delay Tolerant Networking, and Optical Communications.

After observing that CubeSats lacked megabit capacity, possible antennas for the higher frequency, S-band, were studied. To support the communication link range for CubeSats deployable directional antennas such as the log periodic, horn, helical, and patch were considered. A deployable hemispherical tapered helical antenna was then tested for link range against a commercial helical, short/long dipoles, and patch antennas.

Once successful operation of a CubeSat on the S-band for a 2,000 km range was simulated, analysis of LEO constellations followed, to establish that earth observation applications that called upon distributed satellites could be supported.

SSRGT and Flower constellations were simulated in STK and exported to NS-2 to evaluate the topologies for network metrics. Goodput was observed for DTN protocols in the RF S-band and optical channels.

Last, the DTN ION implementation was cross-compiled for a viable CubeSat low power development microcontroller board. A DTN testbed was implemented at Nasa Goddard on an ARM-based AT91SAM9G development board [76]. The AT91SAM9G is considered a commercial off-the-shelf low-power development system for CubeSat command and data handling. Results showed that the peak throughput for an ARM-based microcontroller was 25.64Mbit with a maximum bundle size of 65Kbytes.

Future Work

The work described in Chapters 3 to 6 enables DTN protocols to be applied to CubeSat topologies and platform. However, much research can be done to improve upon experimental results. For example, studies can be performed for antennas in other frequency bands, constellation applications, network parameters, routing protocols, standardization incorporated in network simulators, and improvements to network emulation, DTN implementation, and memory access. The following details further work to build upon the topics covered in Chapters 3 to 6.

Antennas and Radio Frequency Bands

The experiments in Chapter 3 described a deployable helical in only the S-band. Future work involves investigating other antenna solutions packaged for the CubeSat platform. New radio frequency antenna solutions could be developed for the X-band, and Ka-bands. This would allow for increased bandwidth. However, as mesh sensor networks of CubeSats are realized, the issue of capacity and radio interference will

arise. Thus, it is important to design an optical communication payload for CubeSats with a highly accurate pointing system.

Constellation Applications, Networking Parameters, and Routing Algorithms

To build on the topology and networking protocol optimization work of Chapter 4, different orbital mission applications (LEO, MEO, and GEO) could be analyzed for throughput, and compared with and without applying DTN, in network simulations. Chapter 4 described orbits for Earth observation, remote sensing, communications, weather monitoring, and experimental GPS applications. However, other applications (synthetic aperture radar inward to Earth and outward towards space, magnetic fields in the magnetosphere, and upper atmosphere monitoring) for different constellation orbits could be designed in STK and exported to a network simulator.

With new networking simulations, communication protocol parameters (such as slot time, frame spacing, and bundle size if using DTN) could be optimized for a wide array of distributed satellite applications. The current DTN routing algorithms (Epidemic, Prophet, MaxProp, Rapid, and Spray and Wait) are only simulated for terrestrial topologies. Thus, new routing algorithms must be designed for the corresponding network topologies and protocol parameters.

DTN Standardization for Network Simulators

Once these networking parameters and routing algorithms are set, CubeSat transmitters could be “smart, cognitive radios” which sense the current network topology, and transmit with the corresponding protocol standards and routing algorithms. For international agency space craft to receive the signal from these new protocols, intersatellite communication parameters need international standardization. Standardization could be pursued in conjunction with the CCSDS.

Then, to further the DTN networks experiments from Chapter 5, new DTN standards should be incorporated in network simulators such as NS-3, OPNET, and OMNET. In addition, these simulators currently assume 2.4 GHz carrier signals, so new propagation models can incorporate other radio frequencies (X-band, and Ka-band). These simulators could also be modified to support three-dimensional modeling, and orbital mobility models.

Improvements to Network Emulation, DTN Implementations, and Memory Access

To improve DTN network emulation described in Chapter 6, several features could be added. First, for lower resource consumption, the channel emulator software could allow for parallel processing. Second, to more easily experiment, the emulation software could allow for reconfiguration of parameters while simultaneously running. Third, to add to the level of channel emulation realism, the emulator could download real-time atmospheric conditions at ground terminal locations, affecting radio band and optical uplinks and downlinks.

In addition, a more efficient DTN implementation is needed for the lower processing capacities of the CubeSat platform. As described in Chapter 6, the ION DTN implementation maximizes CPU consumption for an ARM-based processor at 65 Kbyte bundle size. Thus, a new lightweight implementation will likely improve CPU usage efficiency.

Last, space wire bus speed on satellite hardware is limited to 400 Mbit/second. This does not allow for high speed memory access need for fast storing and forwarding of gigabyte sized bundles. A possible solution would be to utilize Field-programmable gate arrays (FPGAs), or application-specific integrated circuits (ASICs), as buffers for simultaneous reading and writing of flash memory at rates of 2 Gbits/sec.

LIST OF REFERENCES

- [1] V. Keuren, K. David, "Moon in Their Eyes: Moon Communication Relay at the Naval Research Laboratory, Beyond the Ionosphere: Fifty Years of Satellite Communication" In Butrica, the American Astronomical Society Proceedings, pp. 9–18, Jan 1997.
- [2] V. Lappas, G. Prassinou, A. Baker, and R. Magnuss, "Wireless sensor nodes for small satellite applications," *Antennas and Propagation Magazine, IEEE*, vol. 48, no. 5, pp. 175 –179, Oct. 2006.
- [3] P. Muri, O. Challa, and J. McNair, "Enhancing small satellite communication through effective antenna system design," *Military Communication Conference, MILCOM*, pp.347-352, Nov. 2010.
- [4] D. Israel, and V. Cerf, "IEEE Spectrum Forum," *The Spectrum, IEEE*, vol. 42, no. 10, p. 8, Oct. 2005.
- [5] S. Alluru and J. McNair, "An optical payload for cubesats," in *SmallSat 2010. Connecting the Dots Conference Proceedings*, pp. 1–13, June 2010.
- [6] P. Muri; J. McNair, J. Antoon, A. Gordon-Ross, K. Cason, N. Fitz-Coy, "Topology design and performance analysis for networked earth observing small satellites," *MILITARY COMMUNICATIONS CONFERENCE, 2011 - MILCOM 2011* , vol., no., pp.1940-1945, 7-10 Nov. 2011
- [7] M. Donaldson, P. Anderson, L. Bartamian, *Communication Satellites*, Fifth Edition. AIAA: Aerospace Press Series, Los Angeles, CA, May 2007
- [8] J. L. Stone, Jr. and S. C. Daughtridge, "Satellite exchange in the TDRSS constellation - Techniques and results," in *13th AIAA International Communication Satellite Systems Conference and Exhibit*, D. K. Banks & D. Robson, Ed., pp. 529–536. Oct. 1990
- [9] D. W. Elwell, A. J. Levine, and D. W. Harris, "The tracking and data relay satellite system - An historical perspective," in *AIAA 14th International Communication Satellite Systems Conference and Exhibit*, R. R. Kunath, R. Q. Lee, K. S. Martzaklis, K. A. Shalkhauser, A. N. Downey, & R. Simons, Ed., pp. 588–595, Mar. 1992.
- [10] S. Vaughn and R. Sorace, "Demonstration of the TDRS ka-band transponder," in *MILCOM 2000. 21st Century Military Communications Conference Proceedings*, vol. 2, pp. 1055-1065 vol.2, Nov. 2000.
- [11] S. Pratt, R. Raines, C. Fossa, and M. Temple, "An Operational and Performance Overview of the Iridium Low Earth Orbit Satellite System," *Communications Surveys Tutorials, IEEE*, vol. 2, no. 2, pp. 2 –10, quarter Mar. 1999.

- [12] P. Lemme, S. Glenister, and A. Miller, "Iridium Aeronautical Satellite Communications," *Aerospace and Electronic Systems Magazine*, IEEE, vol. 14, no. 11, pp. 11 –16, Nov 1999.
- [13] C. Fossa, R. Raines, G. Gunsch, and M. Temple, "An Overview of the Iridium Low Earth Orbit Satellite System," in *Aerospace and Electronics Conference. NAECON. Proceedings of the IEEE*, pp. 152-159, Jul. 1998.
- [14] F. Pranajaya and R. Zee, "The Generic Nanosatellite Bus: From Space Astronomy to Formation Flying Demo To Responsive Space," in *Advances in Satellite and Space Communications. SPACOMM. First International Conference on*, pp. 134-140, Jul. 2009.
- [15] B. Klofas and J. Anderson, "A Survey of CubeSat Communication Systems" 5th Annual CubeSat Developers Workshop, Aug. 2008.
- [16] A. Addaim, A. Kherras, and E.-B. Zantou, "Design of a telecommand and telemetry system for use on board a nanosatellite," in *Electronics, Circuits and Systems, 2007. ICEC. 14th IEEE International Conference on*, pp. 455 –458, Dec. 2007.
- [17] R. Diersing, "Characterization of the pacsat file broadcast system," in *Telesystems Conference, 1993. 'Commercial Applications and Dual-Use Technology', Conference Proceedings*, pp. 71 –79, Jun. 1993.
- [18] B. Klofas, "Cubesat Communications Survey Update," in *The summer CubeSat Developers' Workshop, 2011., Aug. 2011.*
- [19] L. Alminde, M. Bisgaard, D. Vinther, T. Viscor, and K. Ostergard, "Educational value and lessons learned from the aaucubesat project," in *Recent Advances in Space Technologies. RAST. International Conference on. Proceedings of*, pp. 57 – 62. Nov. 2003
- [20] K. Schilling, "Design of pico-satellites for education in systems engineering," *Aerospace and Electronic Systems Magazine*, IEEE, vol. 21, no. 7, p. 9, Jul. 2006.
- [21] I. Mas, "A flight-Proven 2.4 GHz ISM Band Satellite," in *Proceedings of the 21st Small Satellite Conference*, Aug. 2007.
- [22] C. Clark, A. Chin, P. Karuza, D. Rumsey, and D. Hinkley, "Cubesat communications transceiver for increased data throughput," in *Aerospace conference, IEEE*, pp. 1-5 Mar. 2009.
- [23] T. Tuli, "Low cost ground station design for nanosatellite missions," in *Proceedings of the AMSAT North American Space Symposium*, 2006.

- [24] N. de Rooij, "Mems for space," in Solid-State Sensors, Actuators and Microsystems Conference, 2009. TRANSDUCERS 2009. International, pp. 17-24 Jun. 2009.
- [25] H. Kayal, F. Baumann, K. Briess, and S. Montenegro, "Beesat: A pico satellite for the on orbit verification of micro wheels," in Recent Advances in Space Technologies. RAST, 3rd International Conference on, pp. 497 –502, Jun 2007.
- [26] G. Minelli, "O/Oreos nanosatellite: A multi-payload Technology Demonstration," in SmallSat 2010. Connecting the Dots Conference Proceedings, pp. 1–13, Jun 2010.
- [27] D. Rowland, "Science of opportunity: Helio-physics on the Fastsat mission and STP-S26," in Aerospace Conference, 2011 IEEE, pp. 1 –12, March 2011.
- [28] G. Crowley, C. S. Fish, G. S. Bust, C. Swenson, A. Barjatya, and M. F. Larsen, "Dynamic Ionosphere Cubesat Experiment (DICE)," AGU Fall Meeting Abstracts, pp. A6, Dec. 2009.
- [29] R. D. Straw, The ARRL Antenna Book. Newington, Conn.: The American Radio League, Inc., 2008.
- [30] J. D. Kraus, Antennas for all Applications. McGraw Hill Book Company, 2001.
- [31] C. Clark, A. Chin, P. Karuza, D. Rumsey, and D. Hinkley, "Cubesat communications transceiver for increased data throughput," in Aerospace conference, IEEE, pp. 1-5, Mar 2009.
- [32] P. Rossoni, "Structural Bus and Release Mechanisms on the ST5 Satellite Summary and Status," in Aerospace Conference, IEEE, pp. 1-14, Mar 2009.
- [33] A. Safaai-Jazi and J. Cardoso, "Radiation characteristics of a spherical helical antenna," Microwaves, Antennas and Propagation, IEEE Proceedings, vol. 143, no. 1, pp. 7-12, Feb 1996.
- [34] H. King and J. Wong, "Characteristics of 1 to 8 wavelength uniform helical antennas," Antennas and Propagation, IEEE Transactions on, vol. 28, no. 2, pp. 291-296, Mar 1980.
- [35] H. Hui, E. Yung, C. Law, Y. Koh, and W. Koh, "Design of a small and low-profile 2 times;2 hemispherical helical antenna array for mobile satellite communications," Antennas and Propagation, IEEE Transactions on, vol. 52, no. 1, pp. 346 – 348, Jan. 2004.
- [36] S. Foo and B. Vassilakis, "Dielectric fortification for wide-beamwidth Patch Arrays," in Antennas and Propagation Society International Symposium, 2008. AP-S. IEEE, pp. 1-4, July 2008.

- [37] J. Wertz, H. Messenger, L. Newman, and G. Smith, Mission geometry; orbit and constellation design and management. Microcosm Press Dordrecht: Kluwer Academic Publishers, El Segundo, CA, 2001.
- [38] H. Bedon, C. Negron, J. Llantoy, C. Nieto, and C. Asma, "Preliminary Internetworking Simulation of the QB50 Cubesat Constellation," in Communications (LATINCOM), IEEE Latin-American Conference on, pp. 1-6, Sept. 2010.
- [39] K. Fall, "The ns Manual" 2011. [Online]. Available: http://www.isi.edu/nsnam/ns/doc/ns_doc.pdf
- [40] International Ground Station (IGS) Network 2013. [Online]. Available: http://landsat.usgs.gov/about_ground_stations.php
- [41] J. Cockwell "EDSN: A Large Swarm of Advanced Yet Very Affordable, COTS-based NanoSats that Enable Multipoint Physics and Open Source Apps" in Proceedings of the 16th Small Satellite Conference, Aug. 2002.
- [42] L. Xiangqun, W. Lu, L. Lixiang, H. Xiaohui, X. Fanjiang, and C. Jing, "Omnet a protocol simulator for satellite networks," in Aerospace Conference, IEEE, pp. 1-9, Mar. 2011.
- [43] STK by Analytical Graphics, Inc. [Online] Available: <http://www.agi.com/>.
- [44] J. Jensen, "Remote sensing of the environment: An earth resource perspective. p. 544," Jan 2000.
- [45] R. Shah, S. Roy, S. Jain, and W. Brunette, "Data mules: Modeling and analysis of a three-tier architecture for sparse sensor networks," Ad Hoc Networks, vol. 1, no. 2-3, pp. 215–233, Jun. 2003.
- [46] V. Salomonson and A. Park, "An overview of the landsat-d project with emphasis on the flight segment," in LARS Symposia, p. 236, Nov. 1979.
- [47] P. Slater, "Survey of multispectral imaging systems for earth observations," Remote Sensing of Environment, vol. 17, no. 1, pp. 85–102, April 1985.
- [48] G.Tyc, et al "The rapid eye mission design," Acta Astronautica, vol. 56, no. 1-2, pp. 213–219, Jun. 2005.
- [49] D. Mortari and M. Wilkins, "Flower constellation set theory. part i: Compatibility and phasing," Aerospace and Electronic Systems, IEEE Transactions on, vol. 44, no. 3, pp. 953–962, Aug. 2008.
- [50] T. Vladimirova, X. Wu, C.P. Bridges, "Space-Based Wireless Sensor Networks: Design Issues", IEEE Aerospace Conference, Big Sky, MT, pp.1-14, Mar. 2008.

- [51] W. D'Amico, M. Lauss, "Wireless Local Area Network Flight Demonstration for High Doppler Conditions", John Hopkins APL Technical Digest, Vol 25, No 4, pp. 335-342A Jun. 2010.
- [52] M. Demmer, "The DTN reference Implementation," presentation at the IETF DTNRG Meeting, March 2005. [Online]. Available: <http://www.dtnrg.org/docs/presentations/IETF62/dtn-impl-ietf-mar05-demmer.pdf>
- [53] S. Burleigh, "Interplanetary Overlay Network: An Implementation of the DTN Bundle Protocol," in Consumer Communications and Networking Conference. CCNC. 4th IEEE, pp. 222-226, Jan. 2007.
- [54] M. Doering, S. Lahde, J. Morgenroth, and L. Wolf, "IBR-DTN: an efficient implementation for embedded systems," in Proceedings of the third ACM workshop on Challenged networks, ser. CHANTS, ACM, pp. 117–120 Jan. 2007.
- [55] A. Keranen, J. Ott, and T. Karkkainen, "The ONE simulator for DTN protocol evaluation," in Proceedings of the 2nd International Conference on Simulation Tools and Techniques, ser. Simutools. ICST, Brussels, Belgium, Belgium: ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), pp. 55:1–55:10, Jan. 2009.
- [56] C.P. Bridges; L. Sauter, P. Palmer, "Formation deployment & separation simulation of multi-satellite Scenarios using SatLauncher," Aerospace Conference, IEEE, pp.1-9, 5-12 Mar. 2011
- [57] A. Cunha, P. Feitas, "Evaluation and updates of the integration platform" Deliverable 7.4, FP7 ICT-2 Networking for Communications Challenged Communities Architecture, Test Beds and Innovative Alliances Contract no: 223994, July 2008.
- [58] S. Burleigh, "Rationale, Scenarios, and Requirements for DTN in Space," Report Concerning Space Data System Standard, CCSDS 743.0-G-1. Green Book. Issue 1. Washington, D.C., Aug. 2012.
- [59] P. Muri and J. McNair, "A Survey of Communication Sub-systems for Intersatellite Linked Systems and CubeSat Missions," Journal of Communications, vol. 7, no. 4, April 2012.
- [60] E. A. Willis, "Downlink synchronization for the lunar laser communications demonstration," in Space Optical Systems and Applications (ICSOS), 2011 International Conference on, pp. 83-87, May 2011.
- [61] H. Kruse and S. Ostermann, "UDP Convergence Layers for the DTN Bundle and LTP Protocols," IRTF Delay Tolerant Networking Research Group, no. 1, May 2009. [Online]. Available: tools.ietf.org/html/draft-irtf-dtnrg-udp-clayer-00

- [62] M. Ramadas, S. Burleigh, and S. Farrell, "Licklider Transmission Protocol - Specification," IRTF Delay Tolerant Networking Research Group, no. 10, June 2008. [Online]. Available: tools.ietf.org/html/draft-irtf-dtnrg-ltp-10
- [63] R. Wang, S. Burleigh, P. Parikh, C.J. Lin, and B. Sun, "Licklider Transmission Protocol (LTP)-Based DTN for Cislunar Communications," *Networking, IEEE/ACM Transactions on*, vol. 19, no. 2, pp. 359 –368, April 2011.
- [64] S. Horan and R. Wang, "Design of a Space Channel Simulator Using Virtual Instrumentation Software," *Instrumentation and Measurement, IEEE Transactions on*, vol. 51, no. 5, pp. 912 – 916, Oct 2002.
- [65] D. Koutsogiannis, S. Diamantopoulos, G. Papastergiou, I. Komnios, A. Aggelis, and N. Peccia, "Experiences from Architecting a DTN Testbed," *Journal of Internet Engineering*, vol. 2, no. 1, pp. 219-229, Dec. 2009.
- [66] DtnBone/grc-dtnbone - Delay Tolerant Networking Research Group (DTNRG), Oct. 2012. [Online]. Available: <http://www.dtnrg.org/wiki/DtnBone>
- [67] J. Schoolcraft and K. Wilson, "Experimental Characterization of Space Optical Communications with DTN Protocols," in *Space Optical Systems and Applications (ICSOS), International Conference on*, pp. 248-252, May 2011.
- [68] R. Beuran, S. Miwa, and Y. Shinoda, "Performance Evaluation of DTN Implementations on a Large-Scale Network Emulation Testbed," in *Proceedings of the seventh ACM international workshop on Challenged networks*, ser. CHANTS, pp. 39–42, April 2012
- [69] W.-B. Pottner, J. Morgenroth, S. Schildt, and L. Wolf, "Performance comparison of DTN bundle Protocol Implementations," in *Proceedings of the 6th ACM workshop on Challenged networks*, ser. CHANTS, pp. 61–64 April 2011
- [70] M. Mathis, J. Semke, J. Mahdavi, and T. Ott, "The Macroscopic Behavior of the TCP Congestion Avoidance Algorithm," *SIGCOMM Comput. Commun.* vol. 27, no. 3, pp. 67–82, Jul. 1997.
- [71] R. Quaale, B. Hindman, B. Engberg, and P. Collier, "Mitigating environmental Effects On Free-Space Laser Communications," in *Aerospace Conference, IEEE*, pp. 1–6, Mar. 2005.
- [72] "Tyvak Intrepid System Boards," March 2013. [Online]. Available: tyvak.com/intrepidsystemboard/
- [73] "The iostat Command, Linux Man Page," March 2013. [Online]. Available: <http://linux.die.net/man/1/iostat>
- [74] S. Narayan and P. Lutui, "Impact on network performance of jumbo-frames on IPV4/IPV6 network infrastructure: An empirical test-bed analysis," in *Internet*

Multimedia Services Architecture and Application(IMSAA) IEEE 4th International Conference on, pp. 1–4, Mar 2010.

- [75] M. Luglio, M. Sanadidi, M. Gerla, and J. Stepanek, “On-board Satellite Split TCP Proxy,” *Selected Areas in Communications*, IEEE Journal on, vol. 22, no. 2, pp. 362–370, Jan 2004.
- [76] P. Muri and J. McNair, “A performance comparison of dtn protocols for high delay optical channels,” in *Wireless Communications and Networking Conference Workshops (WCNCW)*, 2013 IEEE, April 2013.

BIOGRAPHICAL SKETCH

Paul Daniel Muri was born in May of 1985 in Sunrise, FL. Growing up as South Floridian, he was involved in Florida Singing Sons Boychoir, Scouts Troop 366, varsity baseball, and jazz band. Paul graduated from St. Thomas Aquinas High School in Ft. Lauderdale, FL in 2004.

While at the University of Florida, Paul received his BS, MS, and PhD in Electrical Engineering in 2009, 2010, and 2013 respectively. Paul researched in the Wireless and Mobile Systems (WAMS) lab under the advising of Dr. Janise McNair, in the Electrical and Computer Engineering department at the University of Florida. A WAMS lab member since 2008, Paul has researched video communications over a hybrid network, small satellite antenna design, and wavelength division multiplexing local area networks for avionics. For PhD research, Paul focused on delay tolerant networking protocols for satellite communications under a NASA Space Technology Research fellowship with Goddard Space Flight Center.

Paul was President of the University of Florida's student IEEE branch from 2009-2010, an elected representative to the Graduate Student Council from 2010-2011, and a department Curriculum Committee member from 2011-2013. He has worked three internships for Motorola in Fort Lauderdale, FL, two internships for Schlumberger in Houston, Tx, and two internships for NASA Goddard Space Flight Center in Greenbelt, MD. In addition, he is an eagle scout, a brother in Phi Mu Alpha Sinfonia fraternity, a tenor in the a cappella quartet 4Sinfonia, a bass player in the Gainesville funk band Superfish, a HAM radio operator (KJ4YRQ), and a certified Scuba diver. Paul plans to continue working in the satellite space communications industry upon graduation.