Spacecraft-ns3: Spacecraft Discrete-Event Network Simulation

Julianna M. Evans

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science
in
Aerospace Engineering

Jonathan T. Black, Chair
Daniel D. Doyle
Jonathan S. Pitt

May 21, 2020
Blacksburg, Virginia

Keywords: Spacecraft, Network, Simulation
Copyright 2020, Julianna M. Evans

Spacecraft-ns3: Spacecraft Discrete-Event Network Simulation

Julianna M. Evans

(ABSTRACT)

As near-Earth space becomes more populated with large constellations of satellites and research into spacecraft autonomy and disaggregation becomes more prevalent, it will be increasingly important to design effective communication procedures between satellites to efficiently share resources and avoid collisions. Though there have been several space networking simulation tools created in recent years, they all lack rigorous astrodynamics models or use high-fidelity but bulky and computationally taxing commercial software. This research presents Spacecraft-ns3, an extension to the ns-3 network simulator. Using a modular approach, Spacecraft-ns3 propagates orbit state, plans discrete events, and analyzes network metrics and flows. A case study using Spacecraft-ns3 is presented for exploratory space network analysis.

Spacecraft-ns3: Spacecraft Discrete-Event Network Simulation

Julianna M. Evans

(GENERAL AUDIENCE ABSTRACT)

Near-Earth space has become more crowded in recent years due to the increasing number of large constellations of satellites in this region. Autonomous vehicle research has been applied to Earth satellites primarily to share power and computing resources between satellites, or to prevent collisions between satellites. Both of these factors require effective communication procedures between satellites, which can be inexpensively simulated with network simulators. However, network simulators are primarily designed for ground-based use, and must be combined with an astrodynamics simulator to effectively simulate satellite networks. This research presents Spacecraft-ns3, an integrated simulator that defines spacecraft orbits and attitude, and analyzes network activity. This simulator improves upon prior simulation efforts by extending the ns-3 network simulator with efficient and high-fidelity astrodynamics models. The Spacecraft-ns3 simulator is demonstrated in an exploratory case study.

Acknowledgments

I would like to thank my advisor Dr. Black for advising my undergraduate and graduate research for the past four years. Thank you to Dr. Doyle for managing my work on the previous Virginia Tech wireless space networking project that this research was built from, and for serving on my committee and providing feedback. Thank you to Dr. Pitt for serving on my committee and providing feedback on this thesis. Thank you to Dr. Joseph Gaeddert, Dr. Louis Beex, and Dr. Thomas Krauss for their prior work on space network simulation that my research was built from. Thank you to the faculty, staff, and students at the Hume Center, Space@VT, and the Aerospace and Ocean Engineering department.

Thank you to my parents, grandparents, and family for your continuous support for me and for my education. Thank you to my friends Sarah, Ha Nuel, Sally, Ashley, Diana, and Leonard for supporting me and bringing so much joy into my life. Thank you to my graduate school sisters-in-engineering Carli, Neslihan, Prachi, and Varakini for the best conversations and wonderful food.

Contents

List of Figures

List of Tables

List of Abbreviations

CSV Comma Separated Values

DTN Delay-Tolerant Network, or Disruption-Tolerant Network

ECI Earth-Centered Intertial

GCRS Geocentric Celestial Reference System

GEO Geostationary Orbit

GPS Global Positioning System

IEEE Institute of Electrical and Electronics Engineers

IP Internet Protocol

ISL Inter-satellite Links

ITU International Telecommunication Union

LAN Local Area Network

LEO Low Earth Orbit

LVLH Local Vertical, Local Horizontal

M&S Modeling and Simulation

MAC Media Access Control

MAN Metropolitan Area Networks

MANET Mobile Ad-hoc Network

NASA National Aeronautics and Space Administration

OLSR Optimized Link State Routing Protocol

OSI Open Systems Interconnection

RF Radio Frequency

SDR Software Defined Radio

STK Systems Tool Kit

SVN Software Virtual Network

TCP Transmission Control Protocol

TLE Two-Line Element

UDP User Datagram Protocol

UHF Ultra High Frequency

VANET Vehicular Ad-hoc Network

VHF Very High Frequency

VVLH Vehicle Velocity, Local Horizontal

YAML YAML Ain't Markup Language

Chapter 1

Introduction

As near-Earth space becomes more populated with large constellations of satellites and research into spacecraft autonomy and disaggregation becomes more prevalent, it will be increasingly important to design effective communication procedures between satellites to efficiently share resources and avoid collisions. The implementation of new communication architectures to support these types of satellite configurations will not be a trivial task, and incremental steps towards implementing them in space will be necessary. Aspects of terrestrial wireless networking have been successfully implemented in space, though many of these are "proof of concept" and are temporary technology demonstrations.

Network simulation is the best method of inexpensively and quickly modeling network behavior, and is extremely useful in parameter studies and optimization applications. Research in space network simulation has been a topic of interest for the last twenty years, and is an important step in evaluating the requirements of space networks.

This chapter introduces space networks and the simulation of space networks, and discusses the motivation behind developing an integrated space network simulator.

1.1 Introduction to Space Networks

Space networking extends traditional terrestrial computer networking to the space environment. Space networking includes both ground-to-space segments and space-to-space segments. As the name implies, ground-to-space segments consist of information exchange between terrestrial ground stations and space platforms, whereas space-to-space segments consist of any information exchange or link between two ore more space platforms.

Terrestrial computer networks have revolutionized how information is shared and distributed through networks via the establishment of set protocols and procedures that mandate how information may be transferred from one device to the next. Any computer or other physical device that sends or receives information in a network is called a “node”. The physical attributes of nodes may widely differ, but all nodes can communicate with each other due to these standardized protocols. Wired networks use Ethernet cables to transfer information between network nodes as electrical signals, whereas wireless networks use radio frequencies (RF) to pass information from transmitters to receivers [1]. Wireless networks have different characteristics than wired networks, including higher bit error rates, higher latency, and smaller bandwidth [2]. These characteristics and their application to the space environment will be described in section 3.2.

Wireless networks can be generally be classified according to their routing methods and their reliance on infrastructure, though there exists many other attributes that define them. Networks can be broadly classified as “single-hop” or “multi-hop”, with a “hop” defined as the number of intermediate point-to-point transmissions used in a relaying a message from the transmitting node to the receiving node. Single-hop networks consist of direct links between the information’s transmitter node to the information’s receiver node, and multi-hop networks consist of indirect links that pass through multiple other nodes on the way to the receiver node from the transmitter node. Networks are also classified based on their reliance on infrastructure. Infrastructure-based networks rely on fixed access points or routers that manage communications between nodes. Infrastructure-less networks have no central administration and consist of mobile nodes that manage their own communications with other nodes in the network [1]. Figure 1.1 shows a general taxonomy for wireless networks, along with some examples of each type.

This thesis will focus on space-to-space segment ad-hoc IEEE 802.11 networks with some added traits of MANETs and VANETs. Details about IEEE 802.11 networks, MANETs, and VANETs are described in chapter 3, and the specific details about the 802.11 network used in the network simulation tool developed for this thesis are described in chapter 4.

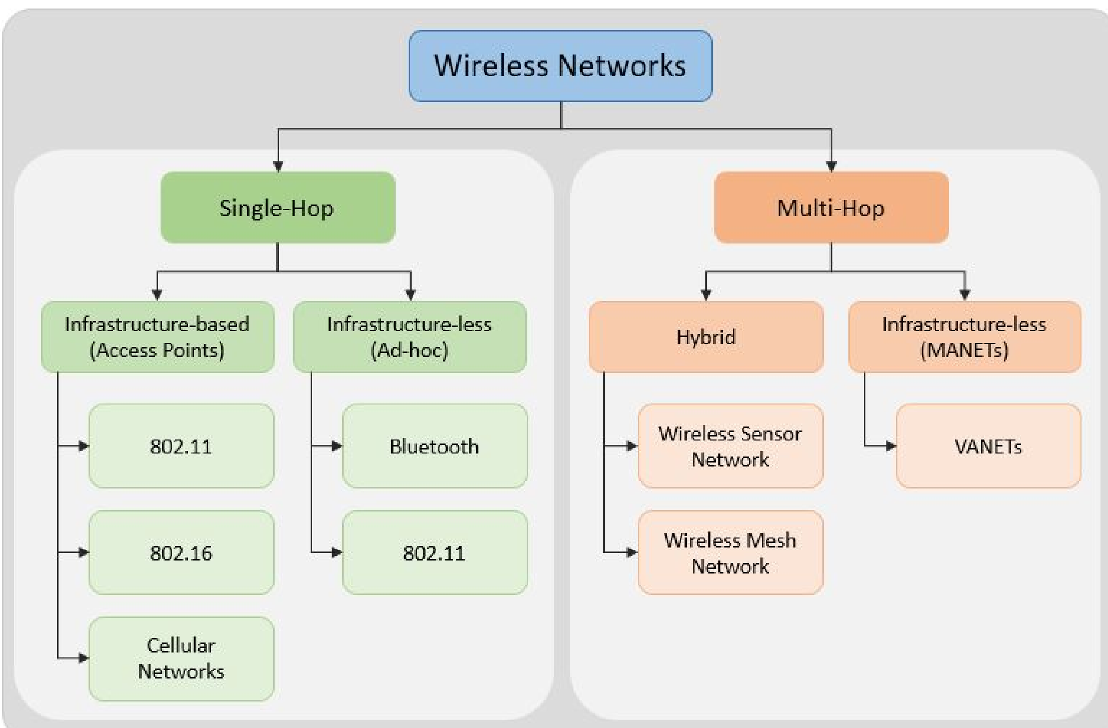

Figure 1.1: **Wireless Network Taxonomy.** Wireless networks can be characterized by their routing methods and reliance on infrastructure. Figure modified from [3].

1.2 Introduction to Space Network Simulation

Space network modeling and simulation consists of the integration of two separate simulation systems—the packet-level network simulation and the astrodynamics simulation. The network simulation is the synthetic model of packet exchange between simulated nodes, and the astrodynamics simulation is the synthetic model of the physics of space vehicles in a simulated real-time environment. There have been many different methods of integrating these two simulation components, which will be further discussed in section 2.2. Typically, the trade-off between computational resources and model fidelity has led to computationally-exhaustive solutions that frequently require supercomputers or simplified but computationally-manageable solutions. Some examples of previous integration attempts are included in section 2.2. This research aims to provide an integrated solution that can be used in computationally-restrictive settings. The following two subsections describe the two components of space network simulation—network simulation and astrodynamics simulation.

Network Simulation

Network simulation is a modeling and simulation technique used to model packet exchange between virtual hosts in a Software Virtual Network (SVN) [4]. SVNs used in network simulations are used to generate network traffic and packets. Network simulation software models all layers of the Open Systems Interconnection (OSI) model, a conceptual model that breaks down telecommunications and computing system attributes and functions into discrete layers. The network behavior can then be analyzed through the manipulation of network parameters, design choices, and environmental conditions [4]. Users can analyze the generated data and perform trade studies and experiments in a controlled environment. Some common network simulation tools used in space networking simulation are ns-3, OPNET, and QualNet [5].

Typical “out-of-the-box” network simulation software will not be adequate for simulating space communications, since they are designed for sea-level applications of wireless LAN or mobile cellular systems [4]. To accurately model space systems, the Physical (PHY) layer of the OSI model must be modified to accurately simulate signal propagation in space, satellite mobility models, satellite dynamics, transmitter and receiver attributes, and antenna patterns [4]. All of these aspects fall under the scope of the astrodynamics simulation, explained further in the next section.

Astrodynamics Simulation

There are several widely-used commercial tools used to model geospatial astrodynamics, including Systems Tool Kit (STK) developed by Analytical Graphics, Inc., General Mission Analysis Tool (GMAT) developed by NASA Goddard, and FreeFlyer developed by AI solutions. Additionally, open-source tools such as SaVi and WinOrbit are also becoming more popular, though they lack the industry validation of more established commercial solutions [6].

Astrodynamics simulators are used to determine the position and velocity time-histories of space vehicles, the physical models of onboard sensor payload and communications components, and environmental effects due to the Earth’s atmosphere [4]. Line-of-sight and propagation delay effects between vehicles are also modeled. However, none of these solutions provide packet-level network statistics that are required for a high-fidelity space network simulation, which necessitates the need for an integrated network and astrodynamics simulator.

1.3 Motivation

Wireless networking with space applications has been a topic of interest to many universities, Virginia Tech included. Many universities have conducted space network simulation research with varying applications—primarily to evaluate new routing protocols and network configurations in the space regime. Ruiz-de Azúa et al. [7] conducted a survey of previous space network simulation integration approaches, which served as the motivation for this project.

Space-based wireless networking simulation is not an incredibly new topic of research, but the approaches used to simulate space-based wireless networks have recently evolved to match the evolution of trends in space constellation architectures. Traditional networks of monolithic geostationary satellites communicating through bent-pipe infrastructures have relied on extremely high-fidelity point-to-point simulations of uplink and downlink communications with little-to-no focus on inter-satellite or crosslink communications. This type of simulation favors heavy-duty commercial solutions, especially since NASA has many legacy systems that combine STK with QualNet. The introduction of SmallSat and CubeSat technologies led to the need to simulate new communication architectures to support crosslinks of large-scale distributed systems. This type of network is often modeled as space-based versions of cellular networks or ground-based sensor networks, which has led to the application of terrestrial internet simulators to the space regime since commercial solutions like STK do not provide wireless networking communication models.

There have been several different approaches to modeling wireless networks in the space regime, which are listed in subsection 2.2.1. The motivating paper for this research was Ruiz-de Azúa et al. [7], which implemented an extension of ns-3 with a custom layer to integrate networking, operational, and physical components of a satellite. Their group has published several papers on this topic, including [7] and [8]. [7] presents the design of an integrated simulator to emulate distributed satellite systems while [8] presents a framework tailored towards Earth-observing autonomous SmallSat networks. However, their methodology is not rigorous in terms of the astrodynamics simulator, since it does not include high-fidelity orbital propagators or attitude modeling.

Upon further literature review, it was discovered that many researchers with networking backgrounds had developed networking protocols and routing strategies that were applied to the space regime using simplistic circular orbit models. The opposite was not true—there were not many instances in which researchers with aerospace backgrounds applied astrodynamics modeling to general networking applications. Though the use of circular orbits can be adequate for certain types of network analysis, this type of simulation is unable to support more realistic multi-plane non-circular satellite configurations such as satellite constellations, swarms, and clusters. Due to this gap in the literature, it was evident that improving the astrodynamics aspect of an integrated space network simulator was necessary. This research focuses on extending networking applications with an astrodynamics model for space-based networks.

1.4 Research Scope

The research presented in this thesis includes a literature review of current space network simulation efforts and the presentation of Spacecraft-ns3, a new high-fidelity application-generic spacecraft network simulation tool. The tool is then applied towards an exploratory next-generation space network case study.

1.4.1 Research Questions

The following research questions were developed to guide the scope of this work:

1. How can the gap between network simulation and astrodynamics simulation be effectively and adequately bridged, without taxing computational resources or suffering from low-fidelity model assumptions?

2. What do expected future space networks look like, and how can this tool provide users with adequate modeling, simulation, and data analysis of these networks?

1.5 Chapter Summaries

Chapter One introduces the topic of space networks and the simulation of space networks. The motivation and scope of this research is given.

Chapter Two defines space networks and discusses the evolution of space network architectures and their implementation in the near-Earth space regime, theoretical research into space-based wireless networks, and current research topics in space networking. The literature associated with the simulation of space networks is discussed, and a survey of current space networking tools is provided.

Chapter Three describes fundamental concepts in orbital mechanics, networking, and simulation that were used in development of the Spacecraft-ns3 tool.

Chapter Four describes the space network simulator that was developed in support of this research, named “Spacecraft-ns3”. This chapter describes the software used, the overall code structure, and the three main modules of the tool—the astrodynamics module, the event planning module, and the space network analysis module.

Chapter Five discusses two applications of an exploratory next-generation space network case study. The case study is a modified version of a subset of SpaceX's broadband megaconstellation Starlink. The applications evaluate different types of network traffic that could be used on satellite train networks, including UDP and TCP intra-plane relays and inter-plane crosslinks.

Chapter Six summarizes this work and discusses future work that could expand the functionality of the Spacecraft-ns3 tool.

Chapter 2

Literature Review

A review of prior and current literature on space networks and the implementation of discrete-event space network simulators was conducted as part of this research. Space networking encompasses a wide variety of current research topics, including SmallSat clustering and formation flying, distributed satellite systems, security in space vehicles communications, and space information networking. This research focuses on space-based wireless networks, and only considers space-to-space links. Ground-to-space networks prioritizing high-volume data links have been much more extensively studied and are not the focus of this work. The first section of this chapter defines space networking and presents the evolution of space network architectures that generates the need for research into space-based networking. The literature regarding implementation of Internet Protocol (IP) components in space is outlined, as well as more recent theoretical research into IP in the space regime. Current research topics are discussed, including the Internet of Space Things, space-based 4G LTE and 5G networks, and megaconstellations.

While several types of network simulation exist and could be applied to the space regime, discrete-event network simulation is overwhelmingly used by academics, NASA, and the commercial sector in simulating space networks. The second section of this chapter summarizes the various methods used to simulate and test space-based wireless networks. Previous implementations of discrete-event space network simulators are listed with commentary on the strengths and weaknesses of each tool, in terms of space networking prioritizing space-to-space communications. As a result of this survey, it was determined all current implementations of discrete-event space network simulators either use bulky and restrictive astrodynamics tools or have very rudimentary low-fidelity astrodynamics models. This survey, along with prior internal research work at Virginia Tech provided a motivation for this research.

2.1 Space Networks

Space networking is a particularly interesting topic of research due to its applicability towards current trends in academic research and in the commercial satellite industry. These trends include satellite disaggregation, formation flying, megaconstellations of SmallSats, and satellite-to-satellite interactions. Satellite-to-satellite interactions will become increasingly important as higher numbers of satellites are launched into congested orbital regions such as LEO, and on-orbit implementations of experimental cluster and formation satellites become more common. Inter-satellite interactions will likely rely heavily on flight-tested communications methods based on RF technologies, especially in the near future.

2.1.1 Definition of Space Networks

As discussed in chapter 1, space networks are an extension of terrestrial computing networks to the space environment. Like terrestrial computing networks, they may have varying components, platform types, or instrumentation; however, like terrestrial computing networks, they have identical routing, protocols, and communication procedures across the network.

The main difference that exists between terrestrial networking and space networking is node mobility. Terrestrial networks often consist of relatively stationary nodes that interact in a relatively small localized environment. An example of this might be a home network of phones and computers that interact within the confines of a house or commercial building. In opposition to this, space networks consist of either stationary or mobile nodes that interact in a much larger environment. An example of a stationary space network might be two GEO satellites with constant line-of-sight, and an example of a mobile space network might be a couple out-of-plane satellites in LEO interacting whenever they have line-of-sight to each other.

2.1.2 Evolution of Space Network Architectures

Space-based networking approaches have changed over the years to support trends in space architectures. The current philosophy of spacecraft development is “agile aerospace”, a philosophy that prioritizes rapid iteration and small design improvements over many iterations [9]. The shift from traditional monolithic satellite architectures towards large constellations of smaller and cheaper satellites has led to advancements in space-based networking, since space-based networking is the communication infrastructure that supports these applications, such as spacecraft cluster and swarm autonomy, resource sharing among network nodes, and data distribution throughout the network.

For the last twenty years, CubeSats and other SmallSats under 500 kilograms have had an increasing presence in near-Earth space [10]. This has occurred in conjunction with the the expansion of the “New Space” industry—a space sector of agile and independent private sector companies that rely on innovation and small budgets to achieve their goals in space [11]. Characterized by a focus on cost reductions, primary emphasis on optimizing operations and incremental development, and a foray into commercial markets, New Space companies use agile aerospace methods and frequently partner with “Old Space” groups (such as NASA, ESA, Boeing, ULA, and Northrop Grumman) [11]. The New Space industry revolution was only possible due to the 1990 Launch Services Purchase Act (which allowed NASA to purchase launch services from commercial companies) and the 2004 Commercial Space Launch Amendments Act (which made private spaceflight legal) [11].

Until about ten years ago, the satellite market was dominated by big corporations and government agencies. Satellite missions had to follow risk-adverse approach to avoid expensive servicing missions in the case of satellite failures [11]. However, due to the expansion of the SmallSat industry-led New Space companies with much smaller budgets, this risk-adverse approach has migrated towards an “innovation-first” approach. Innovations in space networking have been made possible through the rapid iterations of SmallSats using low-cost terrestrial internet components.

2.1.3 Implementation of Internet Protocol (IP) in Space

In 2000, engineers at NASA Goddard used Internet PING packets to successfully communicate with the UoSAT-12 spacecraft—the first instance that a spacecraft assigned an IP address sucessfully communicated with the terrestrial ground stations [12]. Basic networking tasks were conducted, including automatic spacecraft clock synchronization, file transfer, and mail transfer [12]. These tasks were essential in demonstrating automated file store-and-forward capabilities, which are important in Delay-Tolerant Networking. This effort was part of NASA’s Operating missions as Nodes on the Internet (OMNI) project, which was geared towards using standard terrestrial Internet protocols and technologies on space platforms. Previously, NASA had used an internally-developed protocol called Consultative Committee for Space Data Standards (CCSDS), which was used for space-segment communications and did not integrate well with protocols used in terrestrial networks [2]. CCSDS is very similar to terrestrial TCP/IP, though it is much more limited and requires many more steps to get from the satellite to the end user.

In September 2003, a Cisco Internet router (from Cisco Systems, Inc.) was launched into LEO as a technology demonstration of the effectiveness of IP communication in the space environment [13]. The resulting technology demonstration proved that generic commercial networking devices could be used on space platforms, rather than using mission-specific specialized hardware [13]. Ivancic et al. [13] listed several benefits of IP-based technology and hardware on satellites, highlighting the reduced development and design timelines due to the availability of commercial off-the-shelf components and protocols, increased networking capabilities to remote ground stations, and improved interoperability with ground, air, and space systems as active nodes on the Internet.

In 2008, NASA conducted the first successful test of the Delay Tolerant Network architecture (also called "Disruption Tolerant Network") [14]. DTN is a networking architecture based on traditional TCP/IP architectures, with an additional "bundle" layer that stores and relays messages between nodes (see Figure 2.1 below) [15]. DTN uses a "store, carry, and forward" approach that is more tolerant to link disruption than TCP/IP. Research has also been done to mesh the two architectures together in a "TCP-DTN" architecture [16]. DTN was later deployed on the ISS in 2016, and became the first node of NASA's proposed "Solar System Internet" [14], which aims to link already-existing NASA space networks. More recently, in 2018 NASA sent the first successful ground-to-space DTN transmission from a mobile phone in Antarctica to the ISS [14].

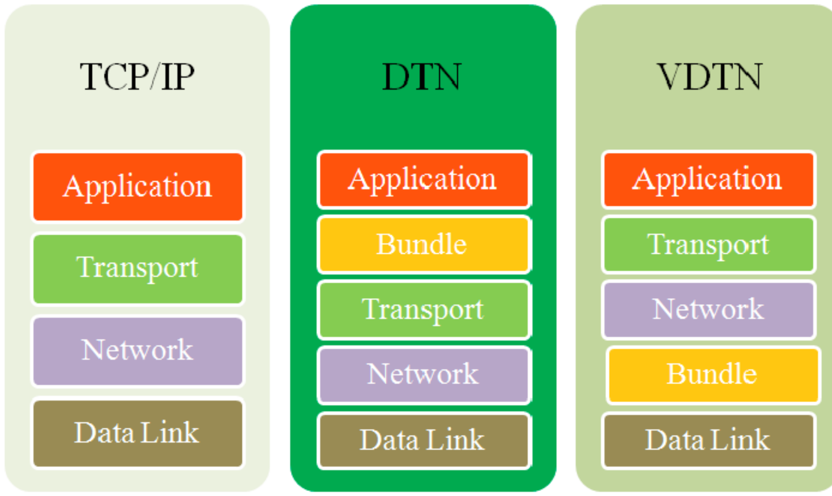

Figure 2.1: **Network Stack Differences Between TCP/IP, DTN, and VDTN.** Delay-Tolerant Networking (DTN) adds an additional "bundle" layer to the TCP/IP stack, which allows networks to temporarily store data and forward it at a later time. Vehicular Delay-Tolerant Networking (VDTN) extends DTN to vehicular ad-hoc networks (VANETs). Figure from [16].

The ISS has been used as a testbed for IP-based technologies, primarily since components can be iteratively tested by astronauts in a much shorter timeline than the timeline required in the design, launch, and operation of a designated spacecraft. In 2010, NASA established a Crew Support Local Area Network (LAN) on the ISS, which provided astronauts remote access to the Internet [14]. Since 2012, the Space Communications and Navigation (SCaN) testbed has been used for on-orbit testing of space networking technologies. Installed on the ISS, its goal is to connect the already-existing Near-Earth Network (NEN) and Deep Space Network (DSN) in a single Solar System Internet (SSI) architecture [17], as well as to support NASA-specific protocols like CCSDS. As a testbed, it comprises of multiple reprogrammable computers and Software-Defined Radios (SDR) that are used to support testbed experiments [17]. This testbed is significant in space networking research since it acts as a technology demonstration for up-and-coming RF research [18]. Experimentation is flexible due to the nature of reprogrammable SDRs.

2.1.4 On-Orbit Implementation vs Theoretical Research

On-orbit implementations of IP technologies have typically been fairly limited due to the cost of implementing new communication architectures in space. Alternatively, theoretical research into space-based IP networks has dramatically increased in the last 10-15 years. Common topics of interest of theoretical research include surveys on antenna components and link budgets, novel constellation designs to support space networks, and new protocols designed to securely transfer information across the space environment with low latency.

The shift towards distributed satellite systems, which cooperatively act together as a single system, has led to many kinds of interesting satellite architectures. Traditional constellations similar to GPS have been well-established in the literature, and have evolved towards the development of more specialized and localized groupings of satellites—including satellite clusters, swarms, trains or "string-of-pearls". Many of these configurations exist primarily in theoretical research, as real-world implementation has typically been "one-off" systems consisting of a single or several SmallSats. As CubeSat technologies have advanced and they become a larger part of the LEO satellite population, they have become the focus of research into near-Earth space internet. In 2015, Rice [19] proposed the use of CubeSats to provide internet services to remote terrestrial locations, as LEO satellite relays for ground terminals, ad-hoc satellites swarms, as LEO-GEO links. SmallSat research has consistently focused on links between space and the ground, and much less so on space-based networks consisting primarily of inter-satellite links or crosslinks.

Theoretical research into link analysis of new space architectures is very common in the literature. Alena et al. [20] evaluated several development paths for space-to-ground and space-to-space SmallSat communications standards. They considered several standards, including the International Telecommunication Union (ITU) Wideband Code Division Multiple

Access (WCDMA) 3G cell phone standard, the 802.11 wireless network standard, and the IEEE 802.15.4 Personal Area Network standard. Their technology assessment included a link margin analysis at the Physical (PHY) and Media Access Control (MAC) layers of the OSI model (see section 3.2), and derived qualitative figures of merit which were used to objectively compare the standards. They found that in general, the WCDMA standard works best for missions requiring long ranges and low data rates, and the 802.11 wireless network works best for closely coupled satellite clusters [20] and high data rates. The 802.11 standard can support more topologies than WCDMA, though it requires adjustments to both PHY and MAC layers. The researchers also expressed interest in the use of cell phone technology for implementing space-to-ground links. This type of theoretical research is essential in the path towards establishing successful space-based networks.

The majority of communications satellites have functioned with a “bent-pipe” concept of operations, in which the satellite link is a single ground-to-space uplink and a single space-to-ground downlink. Future space network architectures move away from this relatively inflexible communication method towards meshed networks that are more similar to the terrestrial Internet. The main interest of deploying IP in space is that it mirrors terrestrial Internet in its capabilities and flexibility, and could easily extend the scope of terrestrial Internet with the same standards and components we are familiar with [21]. However, there exists several roadblocks to implementing spacecraft as nodes on the Internet—primarily with regard to security. Until dynamic end-to-end networking security measures can be guaranteed, it is unlikely that spacecraft in-orbit will become active nodes on the Internet and progress towards on-orbit implementation of ad hoc networking will likely be slow. In the near future it is more likely that small groups of SmallSats will be tested in closed private networks similar to terrestrial Local Area Networks.

2.1.5 Current Research in Space-Based Networking

As traditional space networking architectures move towards more distributed and flexible solutions, on-orbit space networking implementations remain several years behind theoretical proposals, as one would expect. The CubeSat industry has grown substantially in the last twenty years due to its relatively low financial barrier-to-entry, and remains the top method of demonstrating new space networking concepts. Current topics of interest in space networking research include the Internet of Space Things, space-based 4G LTE and 5G networks, and broadband megaconstellations.

Internet of Space Things (IoST)

In 2019, Akyildiz and Kak [22] introduced the concept of an "Internet of Space Things (IoST)" as an analogue to the terrestrial "Internet of Things (IoT)", a global network consisting of all computing nodes connected to the Internet. It is estimated that by 2022, 45% of all internet traffic will be between IP-enabled "things" rather than between people using IP-enabled computers [23]. Akyildiz and Kak [22] noted the current lack of an integrated and flexible space architecture which provides end-to-end varied services for many applications. In particular, they noted among the existing solutions a reliance on the bent-pipe paradigm, the absence of end-to-end functionality, and rigid infrastructure designs [22]. They promote the use of CubeSats as scalable network infrastructures and as passive and active sensors. They propose the use of Software-Defined Networking (SDN) and Network Function Virtualization (NFV) to manage IoST networks, improve resource utilization, and reduce costs [22].

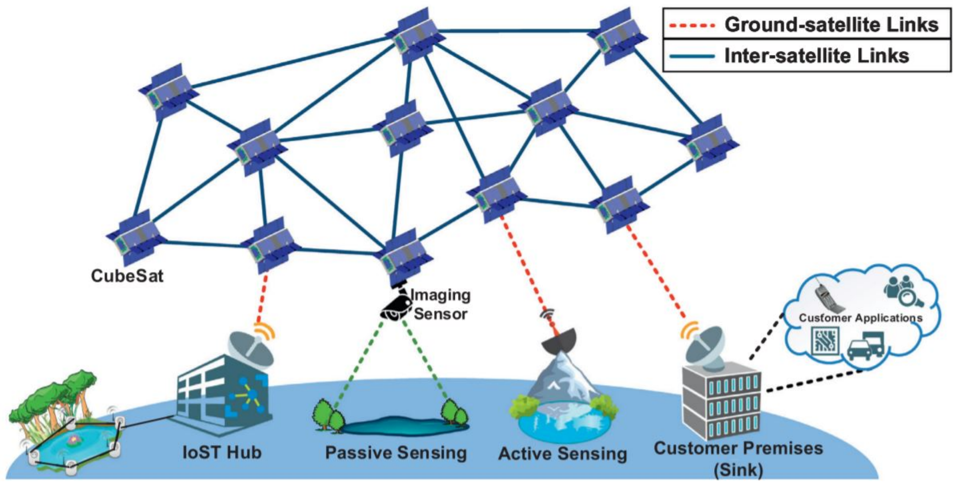

Figure 2.2: **System Overview for Internet of Space Things (IoST) for CubeSats.** CubeSats act as network infrastructure, or as passive and active sensor nodes. Figure from [22].

Space-Based 4G LTE and 5G Networks

Implementation of 4G LTE and 5G networks is an up-and-coming research topic in space networking. "New-space" European companies PTScientist, Nokia, and Vodafone aimed to establish a 4G LTE network on the Moon's surface with a lunar lander and two small rovers [24]. PTScientist led the effort as one of several teams competing in the Google Lunar X Prize competition. The competition was cancelled because none of the leading teams were on track to completing their task before the 2018 deadline, but it remains an important step towards future space networking infrastructure.

Megaconstellations

Many commerical companies have announced plans to launch hundreds or thousands of satellites to create a global space-based broadband internet service. The current company leading this effort is SpaceX, whose StarLink megaconstellation consists of several hundred on-orbit satellites as of this writing. Their goal is to offer terrestrial users internet speeds of 1 Gb/s with latencies of 25-35 ms [25]. They aim to consistently launch batches of satellites using their lower-cost reusable launch vehicles throughout 2019 and 2020, and expect to be able to begin broadband internet service by the end of 2020 [26]. SpaceX has obtained permission from the U.S Federal Communications Commission (FCC) to launch 12,000 satellites, which vastly outnumbers efforts by any other companies [26]. Other companies aim to provide similar services, though none have made as much progress towards the end goal as SpaceX. Some of these companies include OneWeb, Telesat, Amazon, and Samsung, though OneWeb recently filed for bankruptcy in March 2020 due to market turbulence and financial impacts of the COVID-19 global pandemic [27].

Similarly to how SpaceX has dominated the LEO broadband megaconstellation market, Planet (formerly Planet Labs) has dominated the LEO smallsat megaconstellation imagery market. Planet's goal is to use their megaconstellation of 3U CubeSats with telescopic cameras to provide high-definition low-latency Earth-observation imagery to ground-based users. Planet pioneered agile aerospace techniques for large constellations of smallsat imagers, and paved the way for megaconstellations such as Starlink.

Megaconstellations of these sizes will require increased cooperation between nodes to reduce the load of ground-to-space command and control links. Bhasin and Hayden [28] at NASA Glenn looked into the applicability of terrestrial Internet towards NASA programs, listing the following key requirements for post-2010 NASA enterprises: high data rates, high capacity, interactivity with in-space instrumentation, security of operations, real-time data delivery, and seamless interoperability between in-space entities [28]. They recommend the Open Standard Interconnection (OSI) seven-layer model (explained in detail in subsection 3.2.1) to respond to these requirements, citing an expected increase in data handling and simplification of data delivery using this method. The four applications they envision for this application are backbone networks, access networks, inter-spacecraft networks, and proximity networks. Inter-spacecraft networks are the focus of this thesis, as they are most applicable towards semi-autonomous megaconstellations and other envisioned space architectures of the future.

2.1.6 Inter-satellite Links

Future space missions are envisioned to be more complex and increasingly autonomous, which necessitates frequent and reliable inter-satellite communications through Inter-satellite Links (ISL). Particularly in swarming and formation flying applications, space-based networks will need to support dynamic routing, intermediate node management, and autonomous reconfiguration [29].

Smallsats have traditionally been deployed in LEO to test new multi-satellite configurations geared towards autonomous distributed satellite systems. Radhakrishnan et al. [29] evaluated several current and future smallsat missions geared towards this purpose. The important takeaway from Table 2.1 is the reliance on traditional RF solutions for on-orbit ISL research. Optical links using lasers have also been proposed, but they do not have much flight heritage in the context of multi-satellite inter-satellite communications [30], since they have traditionally been used only as technology demonstrations for point-to-point or single-relay communications.

Table 2.1: **Current and Future Multi-Satellite Smallsat Missions.** This table shows several multiple-satellite smallsat missions in operation or planned for the future. These missions show a heavy dependance on RF-based inter-satellite communication systems. Mega-constellations described earlier in the report are not included here. Table from [29].

Mission name	Number of small satellites	Mass of small satellites (Kg)	Inter-satellite links	Inter-satellite communication approach	Launched/Projected launch year
GRACE	2	480	Available	RF based (S-band)	2002
ESSAIM	2	120	Not available	Not available	2004
PRISMA	4	145, 50	Available	RF based (UHF-band)	2010
ELISA	4	130	Not available	Not available	2011
EDSN	8	1.7	Available	RF based (UHF-band)	2015
QB-50	50	2, 3	Available	RF based (S-band)	2016
PROBA-3	2	320, 180	Available	RF based (S-band)	2017
eLISA	3	To be determined	Available	Optical based (LASER)	2028
MAGNAS	28	210, 5	Available	RF based (UHF-band)	To be determined

Yost [31] evaluated flight heritage for RF-based inter-satellite communications for multi-satellite smallsat missions. UHF, VHF, and S-band communications systems have strong flight heritage, whereas X-band through Ka-band is less common but flight proven. This migration towards higher frequency bands is mainly due to crowding in lower RF frequency bands and higher achievable data rates at higher frequencies [31].

2.2 Simulation of Space Networks

As wireless networks have become more complex and nuanced, our ability to simulate them in synthetic environments has become more important in determining their applicability to real-world applications. Simulation remains a fairly cheap and accessible way of testing networking protocols and applications in a controlled environment, which is a particularly attractive option for New Space companies trying to reduce costs. Table 2.2 lists network testing techniques along with their strengths and weaknesses. Simulation and emulation remain the most popular options for studying space networks [5].

As described Table 2.2, network simulation tools often use discrete-event or flow-based models. Of the two, discrete-event simulation is the standard method used in space network simulation. Discrete-event network simulation is more fully described in subsection 3.3.1. In particular to space applications, the network simulator ns-3 is popular in university research due to its open-source nature, and QualNet has been the primary network simulator used NASA enterprises.

Table 2.2: **Network Testing Techniques.** Several methods can be used to test networks. They are listed from most theoretical (Mathematical Models) to most realistic (Field Testing). Table from [5].

	Strengths	**Weaknesses**
Mathematical Models	Quick and cheap means for basic analysis.	Makes assumptions and lacks realism in terms of system capabilities.
Simulation	Possible faster-than-real-time analysis, using discrete-event or flow based models. Still relatively cheap.	Uses simplified model code, not real systems, and includes many assumptions.
Emulation	By using the actual hard/software, emulators can provide accurate understanding of how systems will perform when deployed.	Typically costly because all of the components need to be purchased or written, configured, and maintained.
Large Scale Testbed	Grants researchers the ability to observe how real hosts behave on a realistic network.	These are often shared resources for many different projects and research labs, therefor they are not always readily available to researchers.
Field Testing	Provides the absolute best understanding of how a system will really perform in the wild.	Are expensive and difficult to do at larger scale or with controlled conditions.

2.2.1 Previous Implementations of Discrete-Event Space Network Simulators

Discrete-event network simulators have been used in many space network applications. NASA has several projects geared towards this goal, including Jet Propulsion Laboratory's MACHETE project and SCaN NI&E project, and NASA Glenn's GEMINI and Astrolink projects. These projects all use some combination of the astrodynamics software STK with the network simulator QualNet. A group at UPC BarcelonaTech (Polytechnic University of Catalonia) has used the network simulator ns-3 with a custom extension for modeling satellites and subsystems. A group at Stellensbosch University has developed a custom configuration of Python modules for both aspects of space network simulation. The GLOrbit simulator is an orbital propagator geared towards network topology analysis. All of these implementations will be described in full detail in the following sections.

GEMINI and Astrolink Simulators

The Glenn Environment for Modeling Integrated Network Infrastructure (GEMINI) simulator presents a customized network simulator tool integrated with "astronautical analysis software tools" [32]. They use a dynamic integration environment, which is essentially a parallel process that passes data back and forth between the network simulator (QualNet) and the astronautical physics software (STK) at the physical layer (PHY) of the OSI model.

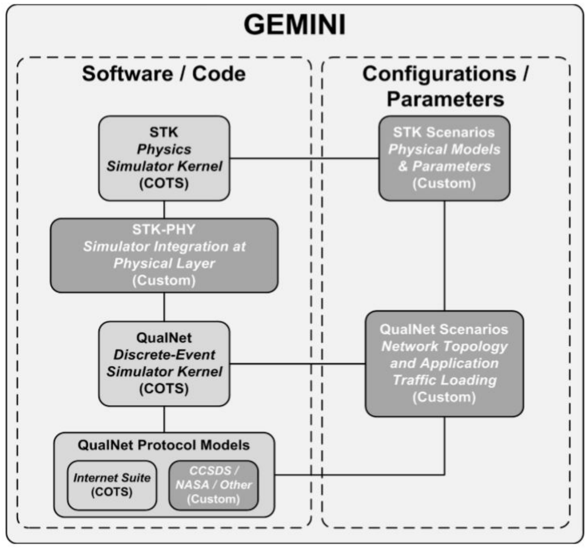

Figure 2.3: **GEMINI Simulator Workflow.** NASA Glenn's GEMINI simulator uses a parallelized approach to integrating a network simulator (QualNet) with an astronautical analysis tool (STK). Figure from [32].

While this integration system does produce high-fidelity results, it is extremely slow and bulky due to the frequent calls of custom interfaces to pass data back-and-forth. The authors iterated on their work, and created Astrolink, a new framework for simulator integration [4]. This simulator addressed the weaknesses of GEMINI and proposed a new integration method using the ns-3 network simulator with STK. They used a caching process to pass data back-and-forth instead of using the STK/Connect interface used in GEMINI, which reduced the number of Remote Procedure Call (RPC) transactions between the two simulators. Though they aim to incorporate many different network simulators and astrodynamics simulators, the work in [4] incorporates only ns-3 and STK and is a prototype simulator.

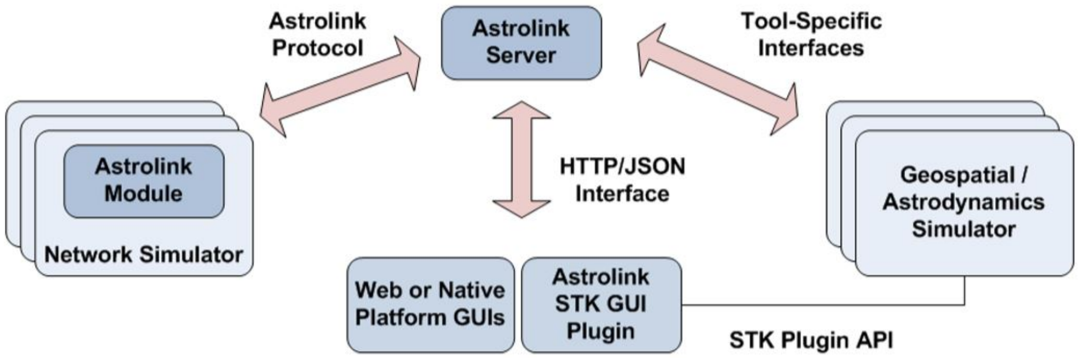

Figure 2.4: **Astrolink Simulator Workflow.** The Astrolink simulator is a server-based simulation framework that uses interfaces to interact with the network simulator and astrodynamics simulator. Figure from [4].

MACHETE Environment

NASA JPL developed the Multi-mission Advanced Communication Hybrid Environment for Test and Evaluation (MACHETE) environment as a collection of simulation tools that can be used interchangeably depending on the requirements of any space network application. It includes many tools, including the Satellite Orbit Analysis Program (SOAP) by Aerospace Corporation, STK by AGI, the Telecommunication Forecaster Predictor (TFP) by JPL, and the Telecomm/Orbital Analysis Tool (TOAST) by JPL [33]. These are all astrodynamics tools that are connected to the QualNet network simulator tool through custom interfaces. QualNet is the simulator used for many NASA tools.

SCaN NI&E Tool

The Space Communications and Navigation (SCaN) NI&E simulator uses a core QualNet engine with added custom protocol models used for space communications [34]. The SCaN Physical model consists of two components: the Tracking and Data Relay Satellite System

(TDRSS) bent-pipe model and a physical propagation model integrated with a Link Budget Library [34]. While this tool can simulate links to a high-fidelity, it only uses a bent-pipe model that is not applicable to the scope of this thesis.

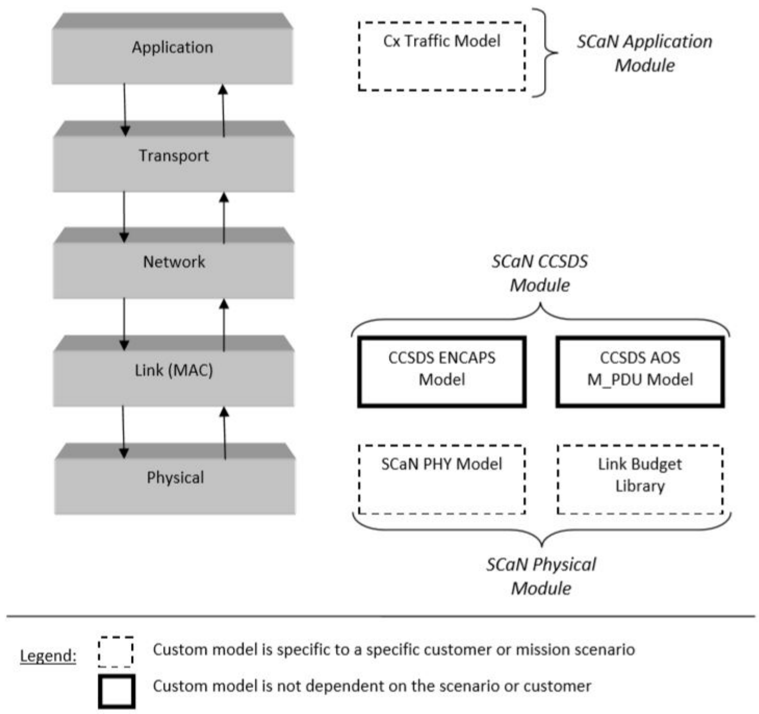

Figure 2.5: **SCaN NI&E Simulator Workflow.** The SCaN NI&E simulator integrates QualNet with several custom modules. Figure from [34].

Integrated Simulator from UPC BarcelonaTech

Araguz et al. [8] from UPC BarcelonaTech presented the concept of a fully integrated simulator combining three different modules that interact with ground and ISL channels: the networking module, operations module, and physical module. They pass user configuration files into a ns-3 engine with an ad-hoc extension to handle modifications to the networking layer, physical layer, and operational layer. This method avoids the parallel or serial methods of passing information from the astrodynamics simulator to the network simulator. Like many other simulators discussed in this chapter, this simulator employs low-fidelity orbital and attitude dynamics. The concept of a custom extension to ns-3 as a collection of modules is the motivation for the Spacecraft-ns3 tool designed for this thesis.

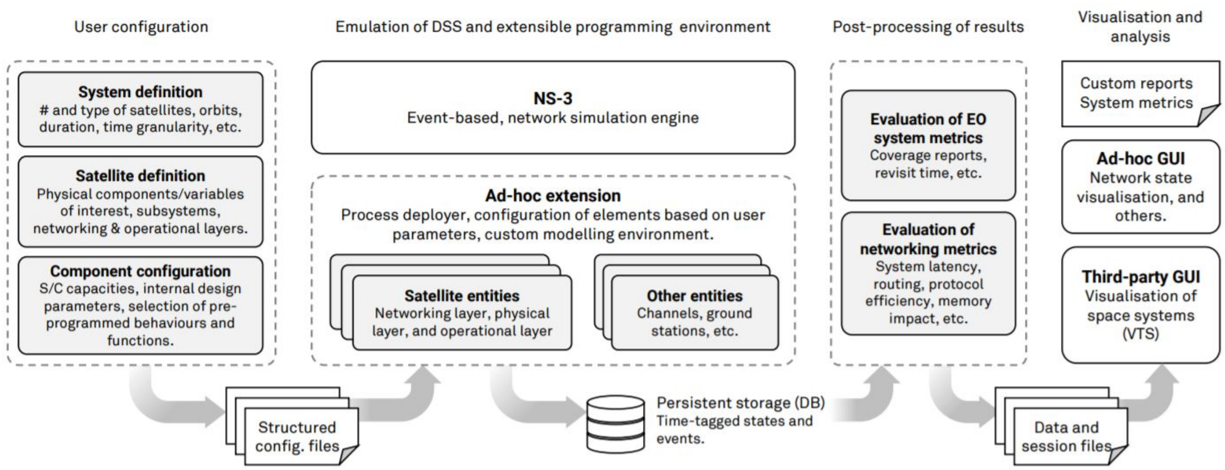

Figure 2.6: **UPC BarcelonaTech Simulator Workflow.** This simulator uses a custom extension from a ns-3 core engine. Figure from [8].

Stellensboch Simulator

Merts and Barnard [35] from Stellensbosch University attempted a fully integrated simulator using a custom combination of Python modules. It uses the SimPy python package as a discrete-event simulator and the PyEphem python package for rudimentary orbital mechanics. Only one satellite-to-ground link is examined, with the satellite orbital mechanics restricted to a 500 km circular orbit over a spherical representation of the Earth [36].

GLOrbit Simulator

Fraire et al. [37] introduced a 3D satellite orbit propagator geared towards network topology analysis for delay-tolerant networks. They describe the physical and topological aspects to the simulator, but do not provide any networking component. The authors report that they are not aware of any commercial or free simulation tools used for packet-oriented switched delay tolerant networks [37]. Due to this, GLOrbit is presented as a topological study of the sparse connections of mobile space networks.

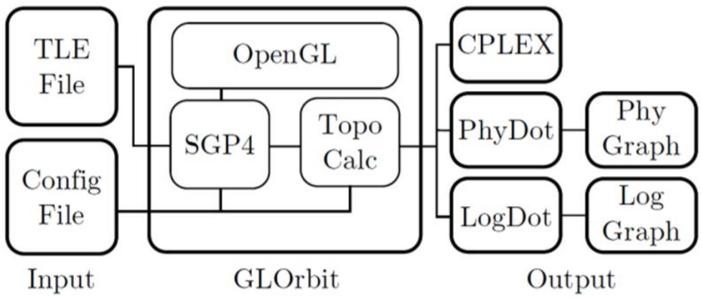

Figure 2.7: **GLOrbit Simulator Workflow.** This simulator architecture describes how GLOrbit propagates orbits. Packet-level network simulation capabilities are not supported. Figure from [37].

Magister Solutions Simulator

Puttonen et al. [38] developed a satellite extension to the ns-3 simulator. While it does include high-fidelity satellite communications modules, it only has one application consisting of a geostationary satellite using a bent-pipe communication structure. This simulation application is useful for traditional satellite communications with a geostationary satellite, but does not apply to any other scenarios involving crosslinks.

2.2.2 Summary of Prior Space Network Simulation Tools

This section summarized current research efforts into space-based networking using discrete-event network simulation. All of the tools that have been described in this section are simulators fulfilling some aspect of the scope of this research. However, to the author's knowledge there is no tool in existence that adequately simulates space networks focused on intersatellite-based configurations. Many existing tools use some combination of STK with QualNet or ns-3, which tends to be bulky and restrictive. Other simulators that do not use STK have very limited applications (i.e. the SCaN NI&E bent-pipe model and Magister Solutions tool) or have very rudimentary astrodynamics models (i.e. the UPC BarcelonaTech and Stellensbosch simulators). This research aims to fill this gap by implementing ns-3 with a high-fidelity astrodynamics model that provides application-generic usage. The choice of using ns-3 and STK as a basis to work from is due to the open-source nature of ns-3 as well as the desire to improve upon previous work. Only a few aspects of astrodynamics models are actually required in this type of simulation, so this research can be considered as a pared-down version of STK with several added custom modules that analyze space-networking functionality.

Table 2.3: **Summary of Simulator Tool Limitations for Space-based Networks.** This table shows a summary of the simulators described in this section, as well as the limitations of each simulator as an integrated tool for space-based networking.

Tool	Description	Simulator Limitations
GEMINI	Parallel interaction between STK and QualNet	Bulky and/or restrictive astrodynamics
Astrolink	Server-based interaction between STK and ns-3	Bulky and/or restrictive astrodynamics
MACHETE	Various astrodynamics tools with QualNet	Bulky and/or restrictive astrodynamics
SCaN NI&E	QualNet with custom addition	Bulky and/or restrictive astrodynamics (uses bent-pipe model only)
UPC BarcelonaTech Simulator	ns-3 with custom addition	Rudimentary astrodynamics
Stellensbosch Simulator	Custom combination of Python packages	Rudimentary astrodynamics
GLOrbit	Custom orbital propagator	No network simulation aspect

2.3 Summary of Current Literature

This chapter discussed prior and current research into space networks and space network simulation. Current trends in advancing SmallSat technologies, satellite architecture distribution, and commercial megaconstellations and have led to an environment in which research in space networking is becoming more essential. Simulating space networks with high-fidelity modeling is necessary to advance space network concepts towards future implementation. New space architectures that rely more heavily on autonomy in distributed satellite constellations will require secure communications systems that handle varying network loads and requirements. In the near term, RF and IP-based solutions remain a flight-proven and reliable method upon which to support future space networks.

This research focuses on space-based network simulation that uses RF and IP-based discrete-event network simulation. Current research topics in space-based networking (subsection 2.1.5) motivated the application of this research, which focuses on satellite megaconstellations. Inter-satellite links (subsection 2.1.6) are the type of communication links that are evaluated in this research. The network simulation work done in this research uses an approach similar to section 2.2.1, in which a custom astrodynamics model extension is applied to the network simulator ns-3.

Chapter 3

Fundamental Concepts for Space Network Simulation

The space network simulation described in this work draws on multiple areas—astrodynamics, wireless networking, and discrete-event simulation. Relevant fundamental concepts in each area are discussed in this chapter. Many of the concepts related to orbital mechanics are involved with the conversion of terrestrial wireless network dynamics to space environment dynamics. The concepts related to wireless networking focus on the network attributes used in the case studies of this research, including wireless network standards typically used in the space environment and characteristics of the 802.11 network analyzed in this research. These characteristics include typical frequency bands used in the space environment, typical RF components used in space communication, and free-space propagation loss. The concept of discrete-event network simulation is briefly discussed, as well as its application towards space networking.

3.1 Orbital Mechanics for Space Networking

Converting a network simulator designed for terrestrial wireless networks requires understanding of orbital mechanics concepts, including coordinate frames, orbit propagation, spacecraft attitude and antenna pointing, and antenna models. This section describes inertial and spacecraft body frames used to simulate spacecraft attitude, the SGP4 orbital propagator, and two types of antenna used in simulations in this work.

3.1.1 Coordinate Frames

Several coordinate frames were used in this research, including an inertial frame and several local orbit frames. Ns-3 currently only supports a Cartesian coordinate system, so satellite position and velocities vector representations must use this frame [39]. Spacecraft-ns-3 uses the Python library *Skyfield* to generate satellite position and velocity time-history data using TLE/SGP4 orbit propagation [40]. Spacecraft-ns3 uses the Python library *poliastro* for Keplerian orbit propagation. Skyfield generates satellite positions relative to Earth's center in the Geocentric Celestial Reference System (GCRS), which is a more precise version of the J2000 coordinate system. The GCRS system is relative to Earth's center of mass rather than the solar system barycenter, so it is especially applicable to near-Earth satellites. This frame is an Earth-centered inertial frame (ECI) with with the $\hat{\boldsymbol{X}}$ axis pointing in the vernal equinox direction, the $\hat{\boldsymbol{Z}}$ axis pointing in the direction of the geographic North Pole, and $\hat{\boldsymbol{Y}}$ completing the right-handed coordinate set.

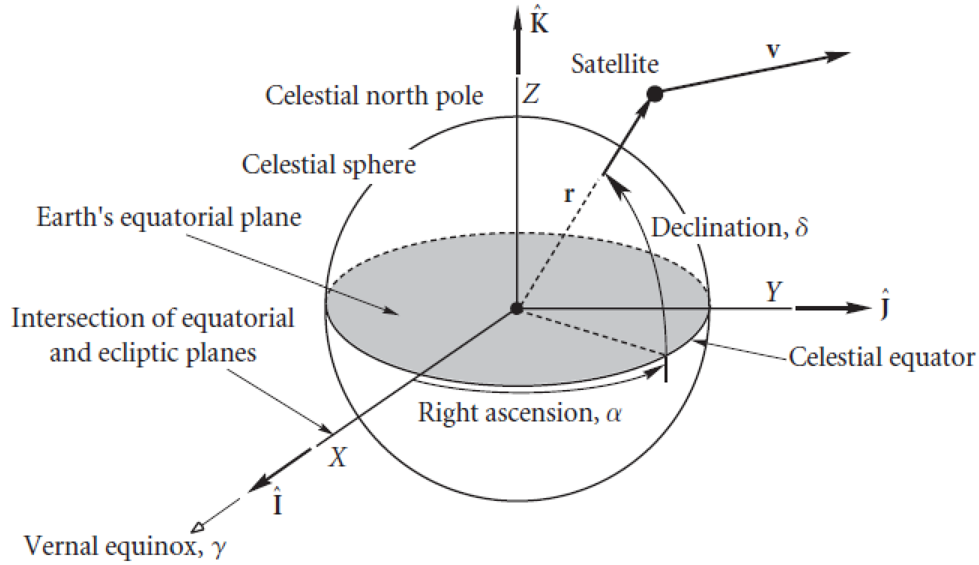

Figure 3.1: **Earth Centered Inertial (ECI) Reference Frame.** The ECI reference frame is an inertial frame used to describe Earth satellite motion. Figure from [41].

Satellite-based orbit coordinate frames were used to describe the attitude of the spacecraft in the local orbit frame. The spacecraft body frame may differ from the orbit coordinate frames; however, in this research the spacecraft attitude is constrained to the orbit frame. Three orbit frames were used in this research: Vehicle Velocity Local Horizontal (VVLH), Local Vertical Local Horizontal (LVLH), and Nadir Alignment with Orbit Normal Constraints. The VVLH frame is equivalent to as the "ECI velocity alignment with nadir constraint" attitude profile in STK. All three of these coordinate frames are based on STK-defined attitude profiles.

VVLH Frame

This frame is also referred to as the "ECI Velocity Alignment with Nadir Constraint" attitude profile in STK. The $\hat{\boldsymbol{Z}}$ axis points in the negative position vector (nadir), the $\hat{\boldsymbol{Y}}$ axis points in the negative orbit normal direction (- $r \times v$), and the $\hat{\boldsymbol{X}}$ axis completes the right-handed coordinate frame.

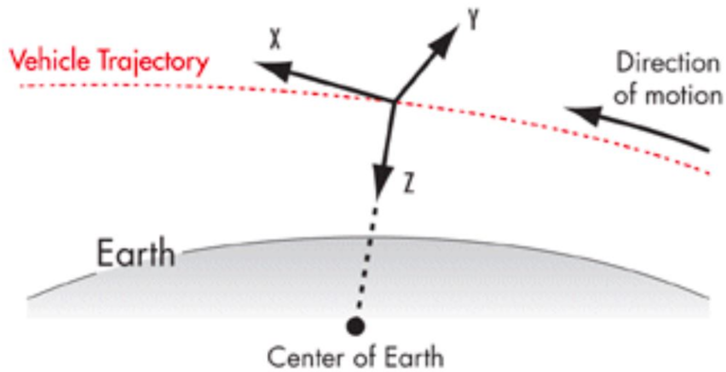

Figure 3.2: **Vehicle Velocity Local Horizontal Frame.** The VVLH frame is characterized by axes in the nadir and negative orbit normal directions. Figure from [42].

LVLH Frame

The Local Vertical, Local Horizontal frame is a common orbit frame with the $\hat{\boldsymbol{X}}$ pointing in the direction of the vehicle position vector, the $\hat{\boldsymbol{Z}}$ axis points in the orbit normal direction, and the $\hat{\boldsymbol{Y}}$ axis completes the right-handed coordinate frame. The LVLH frame is used for the International Space Station to maintain constant orientation with respect to the Earth.

Nadir Alignment with Orbit Normal Constraint

This frame is another local orbit frame with the $\hat{\boldsymbol{X}}$ axis pointing in the negative orbit normal direction, the $\hat{\boldsymbol{Z}}$ axis pointing in the opposite direction to the positive vector (nadir), and the $\hat{\boldsymbol{Y}}$ axis completes the right-handed coordinate set.

3.1.2 Orbit Definition and Propagation

This research evaluates space networks of currently active satellites in near-Earth space, as well as theoretical satellites propagated using classical orbital elements. Classical orbital elements are propagated using Two-Body or Keplerian models, or by using models with some perturbations (such as the J2 and J4 models). The North American Aerospace Defense Command (NORAD) maintains element sets of all resident space objects, which includes functional and non-functional satellites, rocket bodies, and debris. These element sets, called Two-Line Elements (TLE) are provided to users to propagate satellite orbit position and velocity. It is important to note that to maintain accuracy of the propagated orbital state the TLEs must be used with one of the five following prediction models—the SGP, SGP4, SDP4, SGP8, or SDP8 [43]. Some of these models are used for near-Earth satellites (the SGP, SGP4, and SGP8 models) while others are used for deep-space satellites (the SDP4 and SDP8 models) [43]. Spacecraft-ns3 uses the SGP4 model since it is used by the Skyfield orbital propagator library.

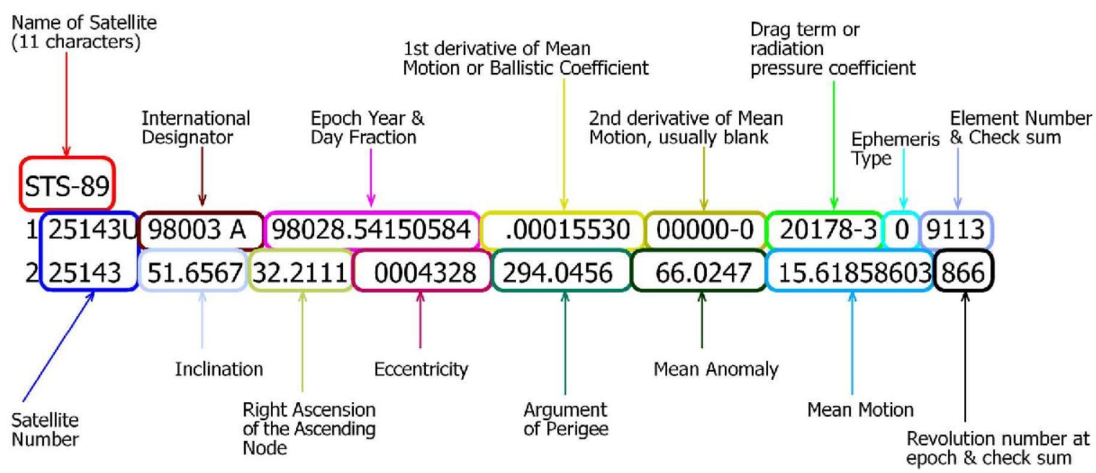

Figure 3.3: **Two Line Element Set.** NORAD provides element sets of resident space objects called Two Line Element (TLE) sets. Figure from [44].

3.1.3 Spacecraft Attitude

The spacecraft body frame attitude is aligned with any of the following orbit frames: VVLH, LVLH, and Nadir Alignment with Orbit Normal Constraint. Since ns-3 operates in a Cartesian coordinate system, the spacecraft attitude must be referenced to this inertial frame. The spacecraft body frame is related to the inertial frame using unit quaternions, which are four-parameter vector representations of a coordinate transformation matrix [45]. The quaternion is assumed to have unit norm

$$q_1^2 + q_2^2 + q_3^2 + q_4^2 = 1 \tag{3.1}$$

The quaternion is represented by several forms

$$q \equiv \begin{bmatrix} \boldsymbol{\nu} \\ s \end{bmatrix} \equiv \begin{bmatrix} -\boldsymbol{u}\sin\alpha/2 \\ \cos\alpha/2 \end{bmatrix} \equiv \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{bmatrix} \tag{3.2}$$

In Equation 3.2, the leftmost representation contains a scalar component (s) with a vector component ($\boldsymbol{\nu}$). The center representation expresses the quaternion in terms of the eigenvector of the rotation matrix (u) and the angle (α) that the initial frame is rotated towards the resultant frame. The rightmost representation is the most commonly used, and consists of the scalar term (q_4) and three vector terms, (q_1, q_2, and q_3). The quaternions may be represented as "vector-first" as shown in the equation above, or they may be represented as "scalar-first", with the scalar component appearing in the q_1 position. Spacecraft-ns3 generates a "vector-first" format as a holdover from previous work done with STK.

The conversion from quaternion to an orthonormal rotation matrix is purely algebraic, and the elements of the matrix are expressed as functions of the quaternion parameters

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} = \begin{bmatrix} q_1^2 - q_2^2 - q_3^2 + q_4^2 & 2(q_1q_2 + q_3q_4) & 2(q_1q_3 - q_2q_4) \\ 2(q_1q_2 - q_3q_4) & -q_1^2 + q_2^2 - q_3^2 + q_4^2 & 2(q_2q_3 + q_1q_4) \\ 2(q_1q_3 + q_2q_4) & 2(q_2q_3 - q_1q_4) & -q_1^2 - q_2^2 + q_3^2 + q_4^2 \end{bmatrix} \tag{3.3}$$

For this research, we are interested in the inverse problem—determining quaternions based on coordinate frame rotations. There have been many methods proposed for this problem, though Shepperd's algorithm [46] remains the most popular, as it is singularity free and uses only one square root term.

In Shepperd's algorithm, Equation 3.1 and Equation 3.3 are used to derive the following equations

$$4q_i^2 = 1 + A_{ii} - A_{jj} - A_{kk} = 1 - trA + 2A_{ii} \tag{3.4}$$

$$4q_4^2 = 1 + A_{11} - A_{22} - A_{33} = 1 - trA + 2trA \tag{3.5}$$

$$4q_iq_j = A_{ij} + A_{ji} \tag{3.6}$$

$$4q_iq_4 = A_{jk} - A_{kj} \tag{3.7}$$

These equations are all the products of two quaternion components. In this notation, i, j, k is a cyclic permutation of $1, 2, 3$, and trA is the trace of A matrix. Shepperd's algorithm then compares the righthand sides of Equation 3.4 and Equation 3.5 to find the largest quantity q_i^2 for $i = 1, 2, 3, 4$. In the case that q_4^2 is larger than any other q_i^2 quantities, q_4 is computed from Equation 3.5 and $q1$, $q2$, and $q3$ are computed from Equation 3.7 to give

$$q_4 = \pm\frac{1}{2}(1 + A_{11} + A_{22} + A_{33})^{\frac{1}{2}} \tag{3.8}$$

$$q_i = (A_{jk} - A_{kj})/4q_4, \text{ for } i = 1, 2, 3 \tag{3.9}$$

Next, quaternion twofold sign ambiguity is checked, which refers to the fact that both the negative quaternion and positive quaternions will be solutions the problem [47]. If q_i for $i \neq 4$ is the largest magnitude quaternion component, it is computed from Equation 3.4, and the other quaternion components are computed from Equation 3.6 and Equation 3.7. The quaternion components are then

$$q_i = \pm\frac{1}{2}(1 + A_{ii} - A_{jj} - A_{kk})^{\frac{1}{2}} \tag{3.10}$$

$$q_j = (A_{ij} + A_{ji})/4q_i \tag{3.11}$$

$$q_k = (A_{ik} + A_{ki})/4q_i \tag{3.12}$$

$$q_4 = (A_{jk} + A_{kj})/4q_i \tag{3.13}$$

As before, i, j, k is a cyclic permutation of $1, 2, 3$. Shepperd's algorithm guarantees a normalized quaternion only if the matrix in Equation 3.3 is precisely orthogonal [47].

3.1.4 Antenna Pointing

Antenna pointing is defined relative to the node's local coordinate system (discussed in subsection 3.1.3), and is centered on the $+X$ axis of the local coordinate system. Spherical coordinates using the ISO convention are used to define the azimuth (ϕ) and elevation (θ) angles of the beam axis of directional antennas. The beam axis correlates to the direction of maximum radiation of the antenna [48]. There are several different antenna models provided in the ns-3 AntennaModel module (see [49]), and this research uses the isotropic antenna model and the bicosine antenna model. Discussion of the isotropic and bicosine antenna models is included in section 3.2.2.

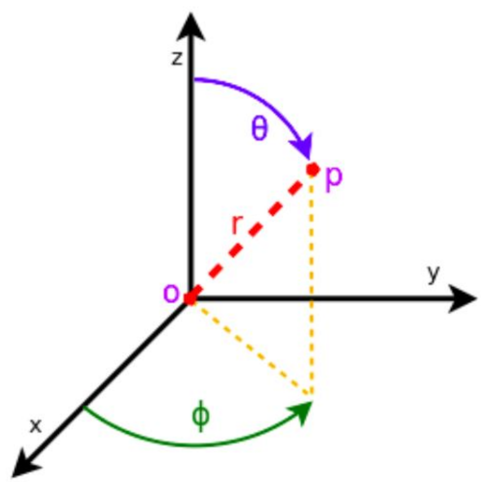

Figure 3.4: **Coordinate Frame for the ns-3 AntennaModel Module.** This coordinate frame describes the azimuth (ϕ) and elevation (θ) angles that define the antenna beam axis relative to the node's local frame $+X$ axis.

3.2 Networking for Space Applications

Space networks overwhelmingly use 802.11 wireless networks in simulation and on-orbit implementation. This section describes 802.11 networks and how they fit into the OSI model, as well as 802.11 network characteristics that are relevant to this research, such as TCP and UDP traffic, OSLR routing, and network metrics that are used in analyzing network performance.

3.2.1 Wireless Networks

Wireless networks are a type of network that allows users to stay connected to the network infrastructure while physically roaming within a predetermined range. This allows for networks such as Mobile Ad-hoc Networks (MANETs). This section describes some background information about wireless networks, the IEEE 802.11 family of standards, routing, and network metrics.

OSI Layer Model

The Open Systems Interconnection (OSI) model is a representation of network functions grouped into seven layers. Each layer provides services and guarantees to the layer above it, and only communicates with the layers directly above and below it [50]. The user interacts directly with Application layer. The Presentation layer converts data formats between Application layer and supports encryption [51]. The Session layer creates, maintains, and destroys communication sessions between devices. The Transport layer segments messages and ensures reliable end-to-end communication using protocols such as TCP [50]. The Network layer delivers data-frames across subnetworks and contains packetized data. The Data Link layer implements communication at the data-frame level. The Physical layer communicates raw bit-streams over a physical medium [50].

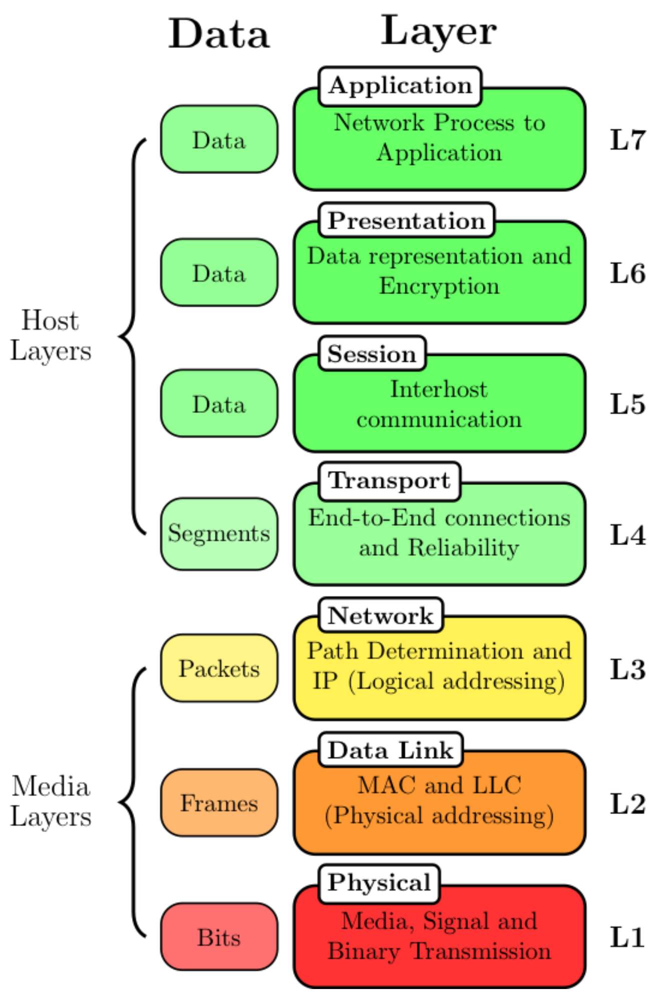

Figure 3.5: **OSI Model.** This figure shows the seven layers of the OSI model. The “Data” column describes the format of the information at each item in the “Layer” column. Figure from [51].

802.11 Networks

The IEEE 802 standards are a family of network standards used in Local Area Networks (LAN) and Metropolitan Area Networks (MAN) [52]. The IEEE 802 standards are restricted to networks that carry variable-size packets, and correspond to the Data Link and Physical layers of the OSI networking model [53].

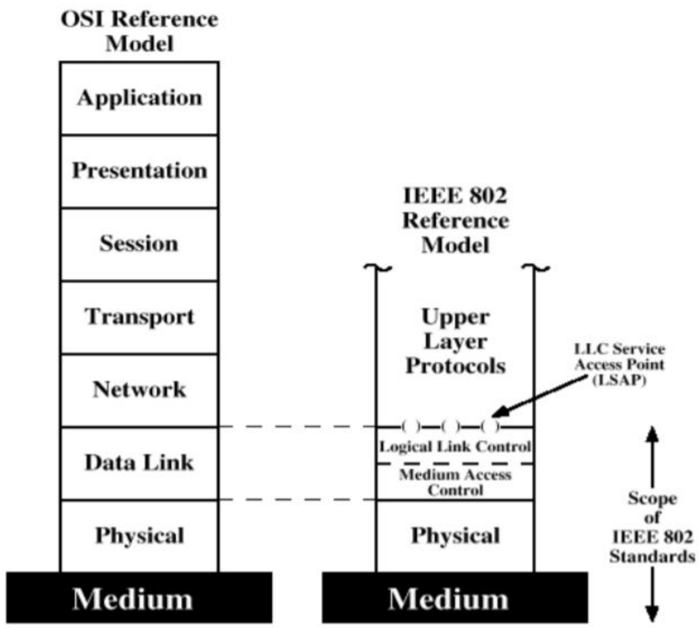

Figure 3.6: **OSI Model and 802.11 Model.** This figure shows how the 802.11 Reference Model fits into the OSI Reference Model. Figure from [54].

The 802 standards of interest to this research are 802.11 a/b/g/n and 802.16 WiMAX. The 802.11 standard is terrestrial wireless network (Wi-Fi or WiFi) designed for indoor multipath environments [55]. In this environment, WiFi can transmit up to 100 meters at 11-54 Mbps for types a, b, and g, and can transmit up to 250 meters at 248 Mbps for type n [56]. The data rate for 802.11n can be improved to 600 Mbps through the use of multiple antennas [55]. It can support point-to-multipoint and ad-hoc configurations.

A similar standard is 802.16-based WiMAX, which can transmit up to 50 km at 280 Mbps in indoor terrestrial environments, and supports point-to-multipoint and mesh configurations [56]. Since WiMAX was designed for long-rage communications and includes a connection-oriented MAC layer, it is not particularly applicable for highly dynamic satellite configurations

The ZigBee standard based on IEEE 802.15.4 has also been used in space networking, though since it has a much more limited range and low data rate than the other standards it has typically been used for intranetworking on a single satellite [57]. The figure below shows some examples of IEEE 802 standards that have been analyzed for space applications.

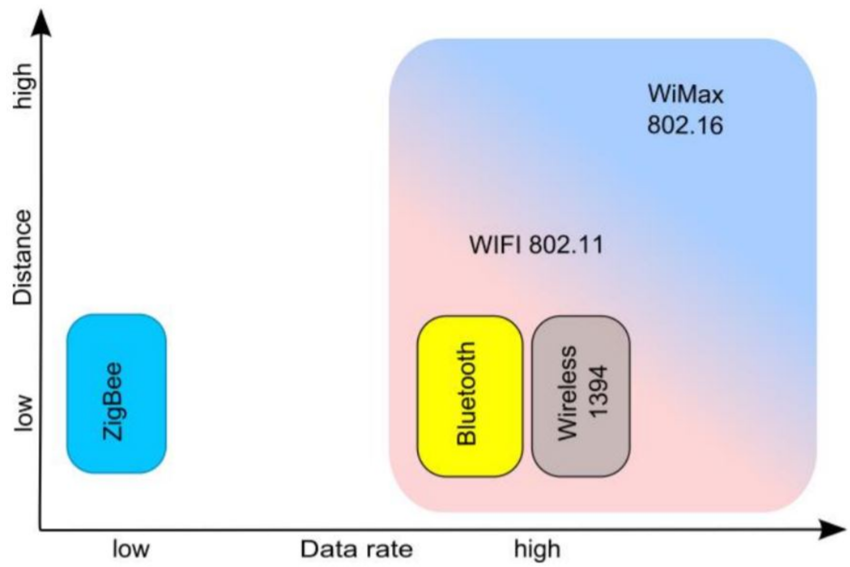

Figure 3.7: **IEEE 802 Standards for Space Networking.** This figure shows several IEEE 802 standards that have been analyzed for space applications. Figure from [55].

TCP/IP and UDP

There are two types of IP traffic evaluated in this research—Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). TCP is also referred to TCP/IP as it is implemented on the IP stack. UDP is also implemented on the IP stack.

TCP communication is much more widely used since it is a connection-oriented protocol, meaning that the transmitter and receiver nodes engage in handshakes and acknowledgements to ensure the entire data stream is sent from the transmitter node to the receiver node. TCP ensures all packets will be received by ordering each packet in the stream and checking the receiver node for an acknowledgement. If the transmitter node does not receive an acknowledgment it will resend the message. This method is reliable due to extensive error checking measures, but adds additional latency and overhead to the messages.

UDP communication is connectionless and does not have extensive error-checking measures. Datagrams are sent in packets to the receiver node in a continuous stream. Messages are not ordered and error-checking is minimal, which does not guarantee reception of the message. UDP is typically used for Voice over IP (VoIP) or streaming applications which rely on fast message delivery.

OSLR Routing Protocol

Ns-3 offers several routing protocols, including AODV, Click, DSDV, DSR, and OLSR [58]. Of these protocols, OLSR is much more favored for ad-hoc wireless network scenarios due to its proactive nature. AODV protocols fit best for spordic data with short data transmissions but does not support constant transmissions and dynamic topology changes well [59]. With OLSR, nodes send periodic Hello messages to discover neighboring nodes, which improves topology awareness across the network [59]. DSDV is also a proactive protocol, but it has a more localized protocol that does not lend itself as well towards large distributed networks and tends to have a large overhead [59].

Networking Metrics

Network metrics are parameters that characterize the behavior of the wireless network. Ns-3 contains Flow Monitor, a network monitoring framework that uses probes to track packet exchange between nodes [60]. Packets are categorized based on the flow that they belong to, and each flow is defined by its source node, destination node, and protocol type. In this research, the protocol used to define flows is either TCP or UDP. The network metrics used to characterize this research are defined below:

- **Latency** Latency is the time required for data to travel from the source node to the destination node. According to Cisco, network latency should not exceed 150 ms for one-way transmission or 300 ms for round-trip transmission [61]. Latency is especially important in TCP communications, as the source node will wait for an acknowledgment of reception from the receiver node before additional data is sent.

- **Jitter** Jitter is an irregular variation in delay that causes disruption in the network. According to Cisco, jitter should be below 30 ms to maintain quality of service [61]. Jitter is more important in interactive applications or protocols, therefore maintaining low jitter is more important in TCP than in UDP. Buffers containing cached data are useful to handle large transmission fluctuations in non-interactive applications.

- **Lost Packets** Lost packets result from a failure to meet requirements in transmission latency. In ns-3, this metric is the total number of packets lost in transmission of a flow.

- **Packet Error Ratio (PER)** The packet error ratio is fraction of packets that are incorrectly received. The fraction of lost packets (packet loss ratio) is a metric used to evaluate the scheduler [62]. According to Cisco, packet loss should not exceed 1% to maintain quality of service.

- **Runtime** In ns-3, the runtime signifies the time taken to complete the flow activity. UDP applications that have an alternating "On-Off" transmission pattern that lasts the duration of the time the node is activated in the discrete-event list. For TCP applications that send data in bulk as fast as possible, the runtime signifies the duration used to send the entire data volume.

- **Throughput** Network throughput is the amount of packets or bit that are successfully transmitted over a period of time. It is typically given in some variation of bits per second, or bps. Latency, jitter, and packet loss all are factors that reduce throughput.

3.2.2 Networking in the Space Environment

Networking in the space environment requires discussion of topics in space communications, including common frequency bands, antennas and RF components, and free-space propagation loss. RF components and frequency bands vary depending on the type of spacecraft.

Radio Frequency Bands

The International Telecommunication Union (ITU) and the Institute of Electrical and Electronics Engineers (IEEE) regulate the use of radio frequency in space applications. The frequency bands typically used for spacecraft begin in the VHF region and end around the Ka-band region. SmallSats typically use lower frequency bands in the UHF and VHF range while military, communications, and “deep space” spacecraft typically use higher frequency bands in the L-Ka range.

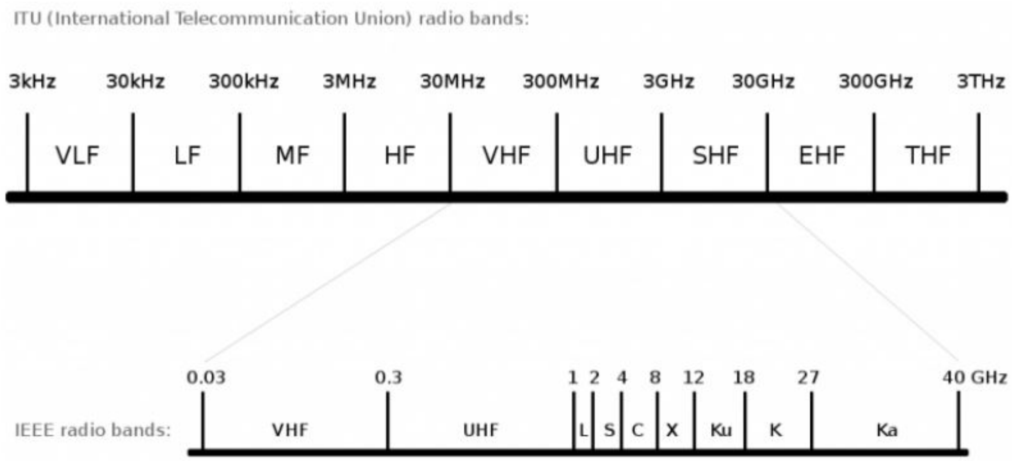

Figure 3.8: **Radio Spectrum Bands Used in Space Applications.** The radio spectrum bands commonly used in space applications are typically within the 30 MHz to 40 GHz range. Figure from [31].

Antennas and RF Components

Transmitters and receivers used to send and receive RF signals vary depending on the type of link needed for the communication. Traditional monolithic-type geostationary communications or Earth-observing satellites use standard highly-directional parabolic reflector antennas. SmallSats typically use monopole antennas, inverted F-shaped antennas, and microstrip patch antennas for telemetry and control links in the UHF and VHF bands [63]. These antennas are particularly useful since they have broad-beam radiation patterns that reduce the need for high-accuracy attitude control. SmallSat payload data links require higher gain antennas, and often use omnidirectional or broad-beam antennas as backup links [63]. Some research has been conducted into specialized antennas for SmallSat intersatellite links, which typically focuses on retro-directive antennas and reflector antennas [63].

There are a couple different antenna models provided in ns-3, including the isotropic model, the parabolic model, and the cosine model. This research uses the non-directional isotropic model and the directional bicosine model.

The isotropic antenna has a hypothetical lossless radiation pattern that has equal radiation in all directions [48]. Even though such an antenna does not exist in the physical world, it is a good reference for more directional antennas. A semi-isotropic antenna is the omnidirectional antenna, which is isotropic in the azimuthal plane and directional in the elevation plane.

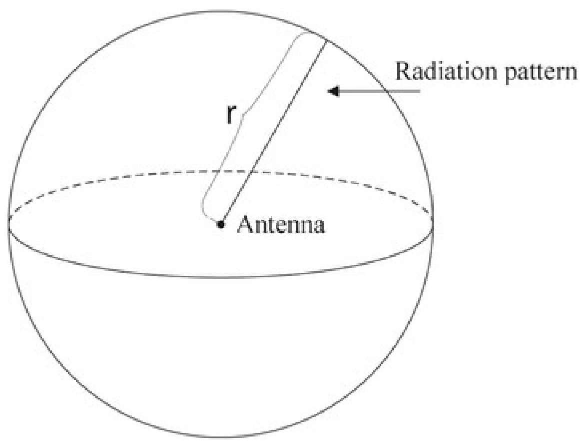

Figure 3.9: **Isotropic Antenna Pattern.** The theoretical isotropic antenna pattern radiates equally in all direction. [64]

The bicosine antenna is a directional antenna with a beam axis referenced to the $+X$ axis of the local coordinate system. It is based off of the cosine antenna pattern. The main lobe of the antenna pattern is centered on the beam axis, and a smaller back lobe is antiparallel to the beam axis. In Spacecraft-ns3, the user can set the availability of the back lobe to "on" (bicosine antenna pattern) or "off" (cosine antenna pattern). The user can set the main lobe beamwidth of the antenna in the azimuth and elevation angles, as well as the offset angles of the beam axis in the azimuth and elevation angles.

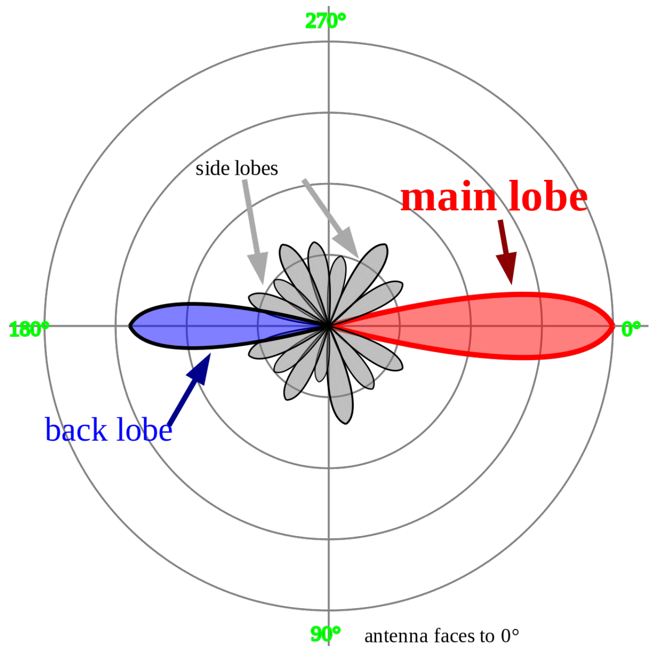

Figure 3.10: **Bicosine Antenna Pattern.** The bicosine antenna has a main lobe in the beam axis direction and a smaller back lobe antiparallel to the main lobe. [65]

Friis Propagation Loss Model

Since the networks of interest in this research are space-based networks that do not consist of any space-to-ground links, the radio path lies generally outside the Earth's atmosphere and is subject only to free-space path loss. Typical ground-to-space links must take into account attenuation, atmospheric absorption, longitude, and latitude [66].

Ns-3 uses the Friis Propagation Model for quadratic path loss in free space. This model is given by Equation 3.14, where P_R is the reception power in Watts, P_T is the transmission power in Watts, G_T is the transmission gain (unitless), G_R is the reception gain (unitless), λ is the wavelength in meters, d is the distance between the transmitter and receiver in meters, and L is the system loss (unitless) [67].

$$P_R = \frac{P_T * G_T * G_R * \lambda^2}{(4 * \pi * d)^2 * L} \tag{3.14}$$

The LEO space regime is partly located in the thermosphere, a region 80-550 km above the Earth's surface [68]. Located in the thermosphere is the ionosphere, a region in which atmospheric particles are broken apart into ions and free-floating electrons by solar radiation [68]. RF signal propagated through this region suffers from signal alteration or degradation depending on the ionospheric activity of the region. Though these atmospheric events do impact the LEO environment, they are beyond the scope of this research and are not considered.

3.3 Simulating Space Networks

Discrete-event network simulation has been an extremely popular way to simulate space networks, given the predictable nature of spacecraft mobility and intersatellite links. Since the future positions of each spacecraft are generally known, we can determine the time intervals in which ad-hoc wireless networks would be successful.

3.3.1 Discrete-Event Network Simulation

Discrete-event simulation is appropriate for interrelated entities which only change their state at discrete points in time from the behavior of any of the simulation entities [69]. These discrete points of state-change are called "events".

Discrete-event simulation is an event-oriented paradigm, which differs from the more traditional continuous paradigm [70]. The continuous paradigm consists of evaluating state variables of each simulation object throughout time. The event-oriented paradigm assumes a constant state of each simulation object throughout time unless a state change is introduced via an event. As seen in Figure 3.11, there are only as many state changes as there are events. The events trigger simulation activity and the state variables of the simulation objects are evaluated.

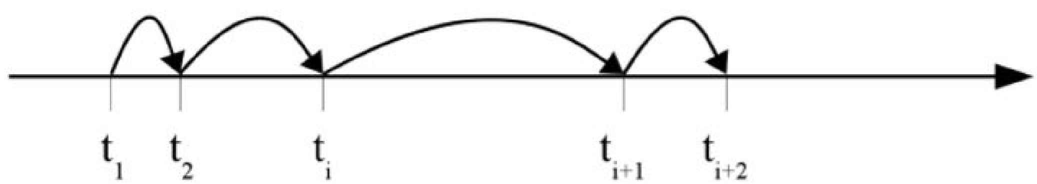

Figure 3.11: **Discrete-Event Simulation Timeline**. This timeline shows the principle of discrete-event simulation. The simulation progression is event-based rather than linear temporal. Figure from [70].

Discrete-event network simulation is the most common method to simulate wireless space networks. Spacecraft collaborating in a network can be represented as a system of connected and attributed entities that change state at discrete points in time [69]. Orbital motion is generally predictable and spacecraft attitude changes and network events occur at discrete times. Space networks consist of opportunistic network activity that is determined through satellite-to-satellite line-of-sight and ranging constraints, which makes it a very suitable application for discrete-event network simulation.

Discrete-event network simulation focuses on analysis of packet exchange between virtual hosts in a network. Probes are installed into network nodes, which then track packet exchange between nodes and generate network statistics. In this research, the ns-3 flow monitor is used to track network statistics. Ns-3 supports probes for IPv4 and IPv6 networks, which classify packets at four events in time: when a packet is sent, when a packet is forwarded, when a packet is recieved, and when a packet is dropped [60].

3.4 Summary of Fundamental Concepts

This chapter reviewed fundamental concepts of space network simulation that are were used in the development of the Spacecraft-ns3 simulator. The orbital mechanics and spacecraft attitude dynamics discussed in this chapter are relevant in extending a terrestrial network simulator like ns-3 to the space environment. Wireless networking concepts applicable to space networks were discussed, and networking metrics used in chapter 4 and chapter 5 were defined. Consideration of the effect of the space environment on wireless networking was discussed, and applicable radio frequency bands, RF components, and loss models were given. Discrete-event network simulation was discussed, as space networks generally use this kind of simulation.

Chapter 4

Spacecraft-ns3: A spacecraft extension to ns-3

A discrete-event space network simulator was developed as part of this research. This simulator tool, called "Spacecraft-ns3", is an extension to the open-source network simulator ns3. This simulator improves previous space network simulation efforts at Virginia Tech and in the literature. Spacecraft-ns3 contains three modules that extend ns-3 simulator capabilities to space-based networks. The Astrodynamics module reads a custom file of spacecraft TLE data, orbital elements, and attitude profiles to generate time-history state data used in the network simulation. The Event Planning module analyzes the state data generated in the Astrodynamics module and recommends the best simulation timesteps for network activity. The Network Analysis module runs the network simulator and analyzes the resulting data. This chapter includes an overview of Spacecraft-ns3 with a description of the code structure, as well as descriptions of each module.

4.1 Overview and Structure of Spacecraft-ns3

The motivation for Spacecraft-ns3 primarily resulted from previous research into space network. Initial work into developing a space network simulator consisted of an ad-hoc combination of STK, Matlab, and ns-3 with several custom interfaces and non-automated data transfer. Several issues with this approach soon came to light:

- Of the numerous features that STK offers, the only necessary features used in this type of simulation were orbital propagators, attitude determination models, and relative motion data. The project workflow was unnecessarily complex due to custom interfaces that had to be designed to link ns-3 to STK.
- Determination of opportune network activity events was done by estimation. The scenario would be manually reviewed in STK using 2D and 3D visualizations to determine timestamps for network events. This process was fairly unstructured and did not scale easily towards larger sized networks.

A modified and more streamlined workflow is necessary to solve these issues. Spacecraft-ns3 is an extension to ns-3 that removes dependency on STK through the development of an Astrodynamics module that contains orbital propagators, attitude determination, and relative motion components. Selection of opportune network activity events is done with the Event Planning module, which evaluates relative motion data to determine time intervals and distances in which nodes are in-range of each other. The Network Analysis model contains some data visualization tools to provide insight into network performance. The structure of Spacecraft-ns3 is shown below in Figure 4.1. The structure consists of three tiers that describe the user-defined input, the modules, and the resulting output data.

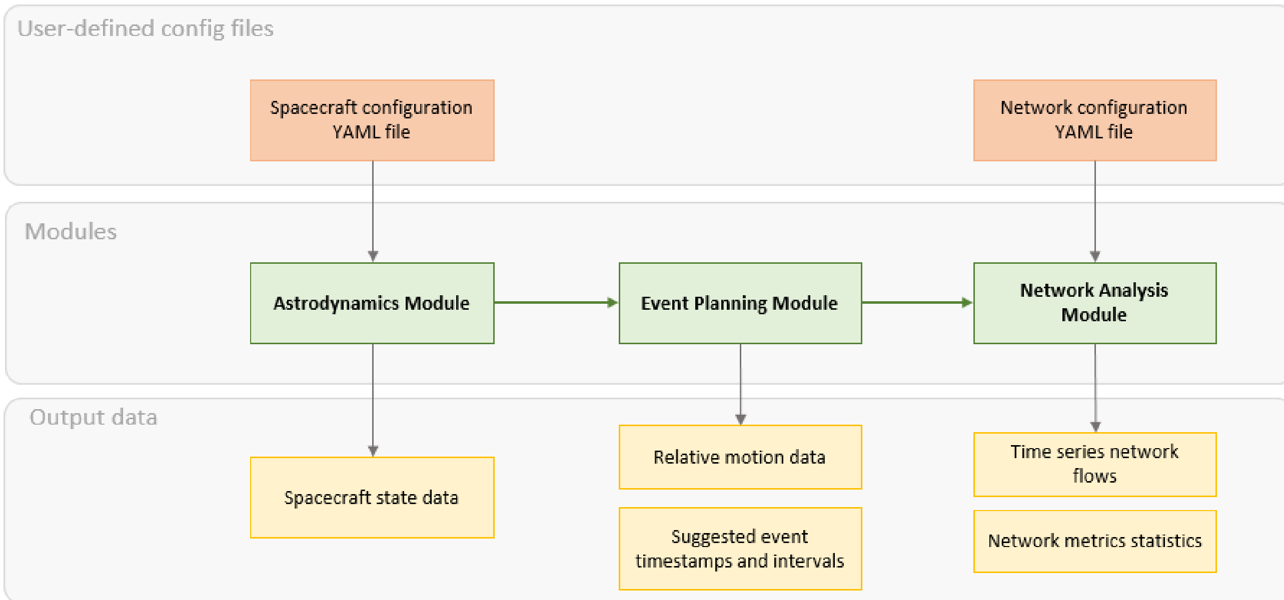

Figure 4.1: **Spacecraft-ns3 Workflow**. The workflow of Spacecraft-ns3 consists of three modules with a three-tiered structure. The top layer includeds user-defined configuration files, the middle layer contains all internal module components, and the bottom layer signifies the data that it output from each module.

4.2 Astrodynamics Module

The Astrodynamics module generates time history state data for each spacecraft object given in the sample input YAML file (example given in Figure 4.2). Users define the time range and interval for the simulation in the top section of the YAML file, and define spacecraft objects in the second section of the YAML file. Each spacecraft object is defined by its name, TLE or orbital elements, and orbit frame that defines the spacecraft body frame attitude.

```yaml
VERSION: 0.1

# This YAML file defines the spacecraft state (position, velocity, attitude)
# and is used to generate CSV files used in the CSV mobility module for ns-3.

# Formatting Notes
#  - datetime format is YYYY-MM-DD hh:mm:ss
#  - attitude frame options are 'VVLH', 'LVLH', and 'Nadir_orbitnormal'. Default is VVLH.

# CASE STUDY 1: LEO Satellite Train


time:
  timeInterval:   1                     # time interval , [s]
  startDatetime:  2020-05-01 08:00:00   # UTC start time, [datetime format]
  endDatetime:    2020-05-01 09:00:00   # UTC end time  , [datetime format]

spacecraft:
  - name: 'S-1008'
    TLE1: '1 44714U 19074B   20123.58334491 -.00003205  00000-0 -20084-3 0  9995'
    TLE2: '2 44714  53.0019  60.4409 0001516  60.7035 163.5650 15.05585230  1806'
    attitudeFrame: 'VVLH'

  - name: 'S-1009'
    TLE1: '1 44715U 19074C   20123.58334491 -.00005202  00000-0 -33808-3 0  9993'
    TLE2: '2 44715  52.9957  60.4732 0001395  78.1054 200.1497 15.05584254 26825'
    attitudeFrame: 'VVLH'
```

Figure 4.2: **Example Spacecraft Configuration YAML File**. This example configuration YAML file is used in the Astrodynamics module to generate state data.

After the YAML file has been created, the spacecraft objects are propagated using Python libraries. Nodes defined with orbital elements are propagated using Keplerian Two-Body methods, and nodes defined with TLEs are propagated using the Skyfield Python library. Developed by Rhodes [71], Skyfield is a high precision generator for planets and Earth satellites. Skyfield is a python implementation of the SGP4 propagator developed by Vallado and Crawford [72]. The Astrodynamics Module consists of orbital propagation and attitude determination using algorithms modified from [46]. The resulting data is formatted into CSV files containing time-series data of spacecraft Cartesian position vectors, Cartesian velocity vectors, and attitude quaternions relative to the inertial frame.

4.3 Event Planning Module

An innovative aspect of this research is the event planning module, which is used to suggest the best instances in the scenario timeline to enact simulation events. Discrete-event simulators require events to trigger network activity. In mobile ad-hoc networks (i.e. MANETs), it is especially important to have knowledge of the network topology at all times. Previous research has typically relied on exported STK data files for time interval data of successful links, which is inflexible and can be computationally costly for large-scale simulations [4]. Additionally, mobile ad-hoc network simulations have traditionally used random mobility models, particularly the Random Waypoint Model. In this model, nodes move independantly towards randomly-set intermediate destinations (i.e. waypoints) with randomly-set velocities [73]. This mobility model is relevant for terrestrial wireless sensor networks for commercial or military purposes, but cannot use event planning methods due to the randomness of the mobility models. Alternatively, non-propulsive satellites have established mobility models that only depend on the Earth's gravitational and atmospheric effects. Due to this, we can have a good understanding of the possible network topology at any given time.

From previous research work at Virginia Tech, it was realized that an event planning module would be very useful for discrete-event satellite simulation. The previous version of mobility models relied on exporting data from STK to a customized version of ns-3, which restricted the astrodynamics analysis to STK. To understand the network topology, the user would review the STK simulation and mark time intervals of interest to use as inputs for the ns-3 simulation. Furthermore, the analysis in STK only focused on the satellite-to-satellite visibility intervals, called "access" intervals. For distance-restricted networks like 802.11, it is important to consider both satellite-to-satellite visibility and relative distance.

The Event Planning module uses the CSV files generated in the Astrodynamics module to compute relative distance and velocity data between each node. Relative distance and velocity data is calculated for each node, which results in two datasets for each node in the scenario. The relative data is then reduced to a list of optimal timestamps based on closest approaches and intervals under a given limit. This type of statistic is relevant for short-interval dynamic passes. Intervals-based statistics list all time intervals in which nodes are within a specified distance of each other. This type of statistic is relevant for determining longest-available communication windows. Examples of this kind of data is is given in Table 5.2, Figure 5.7, and Figure 5.8.

4.4 Space Network Analysis Module

After the Astrodynamics module propagates spacecraft object state data and the Event Planning module recommends opportune timestamps for network activity, the information is passed to ns-3 via a network configuration file. This file is in YAML format, and includes information needed to configure output tracing files, network addressing, radio options, events, and nodes. This configuration format is retained from previous work done at Virginia Tech, though the Event and Node blocks are modified to fit input from the Astrodynamics module and the Event Planning module. Each section of the configuration is described in the following sections.

4.4.1 Network Configuration File

The network configuration file can be broken down into six sections: config, tracing, network, radio, events, and nodes. These sections define the time interval of the scenario, network attributes, discrete events, and node dynamics data. Each section is given below, and full versions of the network configuration file are given in A.

The config section determines the time interval and seed. Ns-3 convention dictates that the scenario time must be in "elapsed time" format, which is reflected in the *startTime* and *runTime* parameters. The *seed* parameter determines the randomness of the network activity. In this research the seed is held constant with a value of "1", which allows for simulation repeatability.

```
VERSION: 0.1

# This YAML file runs the ns-3 scenario using the nodes given in the 'nodes' section
# ns-3 uses the elapsed time convention

config:
  startTime:          2607
  runTime:            500    # [s]
  seed:               1      # random seed for simulation; cannot be zero
```

Figure 4.3: **Example Network Configuration File: Config**. This section of the example network configuration file configures the time and repeatability (seed) parameters.

The tracing section sets parameters associated with the network activity output files. The *flow_stats* and *link_stats* comprise of the point-to-point network activity time series data, which will be discussed in subsection 4.4.2. The *logInterval* is set to 1.0, which sets the time step of the reported data to one second.

```yaml
# configuration parameters for output tracing
tracing:
  prefix:             ./
  animFileName:       sns3-sim.xml
  traceFileName:      sns3-sim.tr
  routeFileName:      sns3-sim.routes
  neighFileName:      sns3-sim.neighbors
  flowFileBase:       flow_stats
  linkFileBase:       link_stats
  logInterval:        1.0
  verboseNet:         false
  verbose:            false
  specAnalyzerNode:   null
```

Figure 4.4: **Example Network Configuration File: Tracing**. This section of the example network configuration file configures the output tracing parameters.

The network section assigns the base addressing scheme, UDP and TCP ports, and the routing protocol. The routing protocol used throughout this research is OLSR, which was described earlier in section 3.2.1. The *HelloInterval* parameter is used to ping neighboring nodes to determine connectivity. The *Willingness* parameter signifies if nodes are willing to carry and forward traffic for other nodes.

```yaml
# network addressing
network:
  addressbase:        10.1.1.0
  bcastaddress:       10.1.1.255
  UDPPort:            7100
  TCPPort:            7200
  routing:            olsr
  routingAttributes:
      HelloInterval:  2
      Willingness:    default
```

Figure 4.5: **Example Network Configuration File: Network**. This section of the example network configuration file configures the base addressing, routing, and port parameters.

The radio section sets parameters associated with the MAC layer, station manager, power levels, and frequency. The MAC layer parameters coordinates channel access across the network, and these parameters are left to default values. The station manager is set to the *Minstrel Wifi Manger* model in ns-3, which is a rate selection algorithm based on acknowledgment feedback [74]. The data mode sets the modulation scheme and data rate (bandwidth) to be used. The transmission power levels is set to 20 dBm with a min-to-max range of 0 dBm to 30 dBm. The base frequency is set to 2412 MHz. The CCA threshold sets the minimum power level of the received signal such that the PHY later declares a busy state. [58].

```
# radio parameters
radio:
  standard:           802.11a
  mac:
      Slot:                   9us     # slot time
      Sifs:                   10us    # SIFS duration
      AckTimeout:             88us    # ACK timeout duration
      CtsTimeout:             88us    # CTS timeout duration
      Rifs:                   2us     # RIFS duration
      BasicBlockAckTimeout: 2us     # Basic Block ACK timeout duration
      CompressedBlockAckTimeout: 112us # Compressed Block ACK timeout duration
  staManager:         ns3::MinstrelWifiManager
  staManagerAttributes:
      UpdateStatistics: 0.1
      LookAroundRate:   10
      EWMA:             75
      PacketLength:     1200
  dataMode:           OfdmRate6Mbps
  txPwrDbm:           20
  txPwrDbmMax:        30
  txPwrDbmMin:        0
  txPwrLevels:        10
  frequencyMHz:       2412
  ccaThreshold:       -62
```

Figure 4.6: **Example Network Configuration File: Radio**. This section of the example network configuration file configures the wireless network, which includes the mac, station manager, and power parameters.

The events section lists all of the events that drive the discrete-event network simulation. The events that drive the simulations in this research are On and Off commands for the WiFi radio, which are used to manage high-volume data transmissions. Hello Intervals transmissions may also trigger events; these events are defined by the start timestamp and duration of the Hello message.

```
events:
  - {type: wifiRadioOn,  address: 10.1.1.10, start: 0}
  - {type: wifiRadioOn,  address: 10.1.1.20, start: 0}
```

Figure 4.7: **Example Network Configuration File: Events**. This section of the example network configuration file configures the event parameters.

The nodes section defines the association between the spacecraft objects defined by the CSV files to the addressing scheme given in the *network* section. Antennas on each spacecraft are defined. The *applications* section defines the receiving node of the transmission. Data may be sent in *telemetry* mode or *data* mode. The data transmission type may either be *OnOff* or *BulkSend*. The *OnOff* data type follows an alternating pattern of "On" and "Off" modes. The "On" mode generates traffic characterized by the given data rate and packet size. The "Off" mode does not generate network traffic. The *BulkSend* type sends data as fast as possible according to the maximum number of bytes that are allowed, which continues until the application is stopped or the transmission completes.

```yaml
nodes:

  - csvfile:          csv_starlink9/S-1009.csv
    address:          10.1.1.20
    antenna: {type: omni}
    applications:
      - {destAddress: 10.1.1.10, class: telemetry, start: 0, size:  250, type: OnOff, protocol: udp, rate: 10kbps}

  - csvfile:          csv_starlink9/S-1008.csv
    txPwrDbmMin:      0
    txPwrDbmMax:      30
    txPwrLevels:      20
    address:          10.1.1.10
    antenna: {type: bicosine, beamwidth-az: 90, beamwidth-el: 90, offset-az: 0, offset-el: 0, backlobe: true}
    applications:
      - {destAddress: 10.1.1.20, class: telemetry, start: 0, size:  200, type: OnOff, protocol: udp, rate: 10kbps}
      - {destAddress: 10.1.1.20, class: download,  start: 0, size:  400, type: BulkSend, bytes: 10e8}
```

Figure 4.8: **Example Network Configuration File: Nodes**. This section of the example network configuration file configures the spacecraft address, CSV data, antenna, and network activity parameters.

4.4.2 Network Analysis

There are several different types of data files generated after the ns-3 simulation runs. Data is categorized by flows from source nodes to destination nodes, or from transmitting nodes to receiving nodes. For each flow, time-series data is given of packets transmitted, packets received, bytes transmitted, and bytes received. Standard network metrics are calculated, including jitter, latency, packet error ratio, runtime, and throughput. Examples of time series network data and network statistics tables can be found in section 5.2 and section 5.3.

4.5 Summary of Spacecraft-ns3

This chapter presented Spacecraft-ns3, a new space networking simulation tool designed for flexible networking implementation using high-fidelity astrodynamics. Spacecraft-ns3 is an extension to ns-3, and does not require any interfacing with astrodynamics tools like

STK. This chapter provides and overview of the structure of Spacecraft-ns3 as well as some examples of configuration files used to run the simulator.

Chapter 5

Spacecraft-ns3 Case Study

This research focuses on space networks consisting of spacecraft-to-spacecraft communications. As discussed in subsection 2.1.4, progress towards implementing on-orbit IP-based networks exists mainly in theoretical research since closely-situated formation flying satellite configurations are not currently implemented in space. Wireless technologies support inter-satellite links of several hundred meters, and thus it remains a topic of theoretical research. However, rather than simulating purely theoretical satellite architectures, we can use current trends to drive theoretical simulation. Current advancements with megaconstellations like Starlink can be blended with theoretical research concepts to provide novel but generally realistic scenarios. The case study evaluated in this research presents a modified version of a subset of the Starlink megaconstellation. SpaceX launched its first batch (Block V1.0) of 60 satellites in May 2019, which were scheduled to support Ka-band communications and optical crosslinks, though both of these elements were dropped from the design [75]. This case study blends a modified version of Starlink with a wireless network to explore an alternate concept of an on-orbit satellite constellation.

5.1 Case Study Motivation

The satellites evaluated in this case study are modelled after a subset of nine of the 60 Starlink Block V1.0 satellites. This subset consists of a train of eight satellites and one satellite in a secondary plane. The satellite in the secondary plane intersects the other eight satellites in the polar regions. The Block V1.0 satellites typically have a 550 km x 550 km orbit with an inclination of 53 degrees [75], however many of the satellites launched after Block V1.0 are grouped into planes of varying inclination.

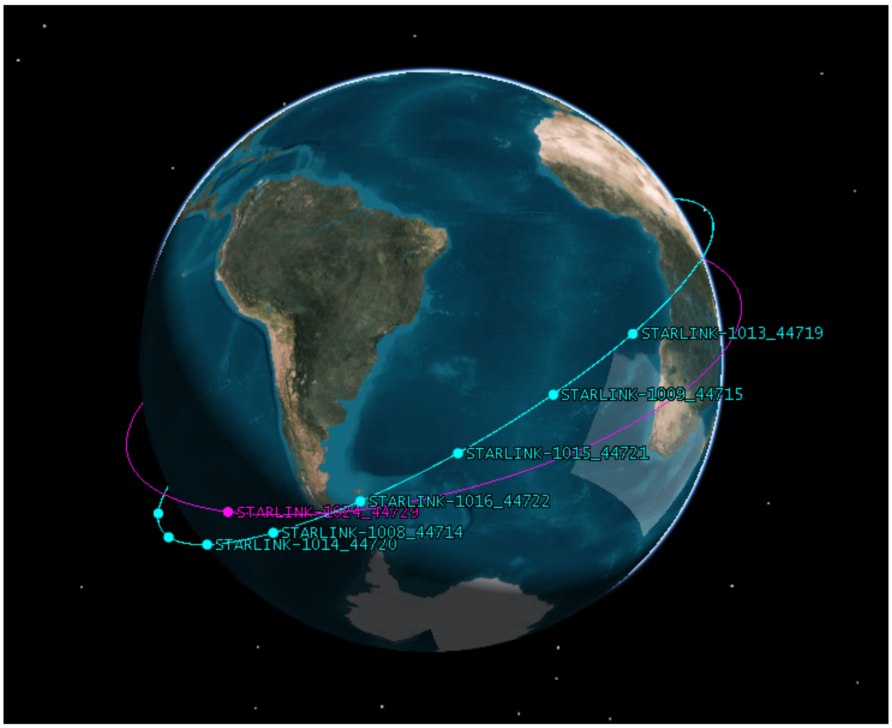

Figure 5.1: **Starlink Block V1.0 Subset**. The case study evaluated in this research is modified from a subset of the Block V1.0 Starlink Satellites. Figure generated from STK.

This case study uses a similar satellite train to the subset of Starlink described above, though the angular spacing of nodes throughout the orbital plane is greatly reduced to support wireless ranging. Furthermore, instead of evaluating a satellite train of eight satellites with one satellite in a separate plane, the modified case study evaluates two sets of satellite trains—four equally spaced satellites in the first and three unequally spaced satellites in the second. This modification was to allow a wider range of networking applications using the same configuration.

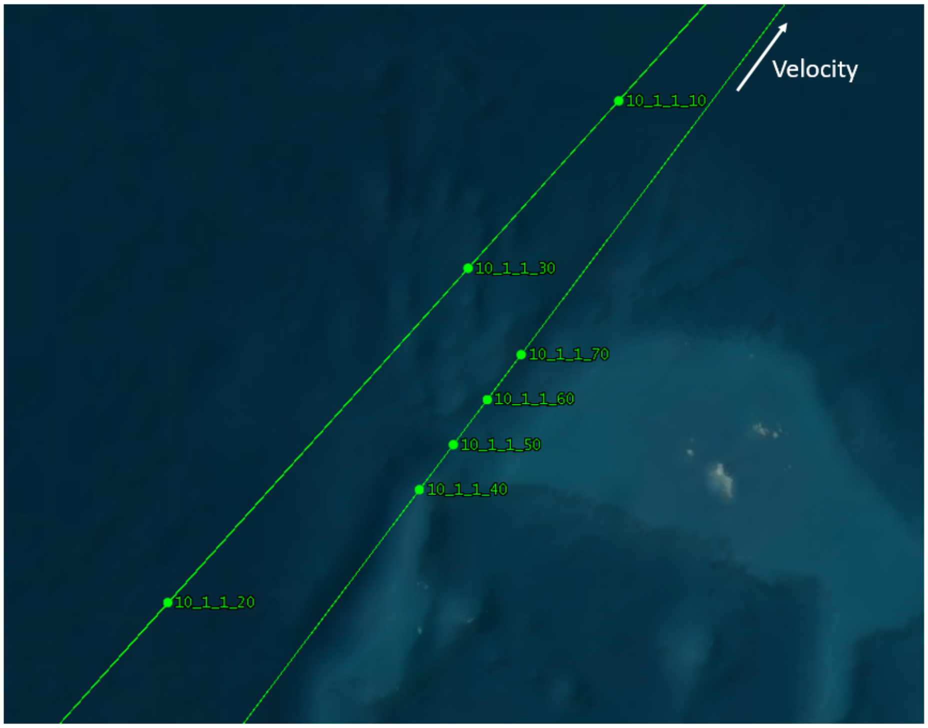

Figure 5.2: **Dual-Plane Satellite Train**. This case study presents a version of Starlink modified to accommodate wireless networking ranges. Figure generated from STK.

5.2 Application 1: TCP Relay and UDP Telemetry

The first application of the dual-plane satellite train case study evaluates four events separated across a ten minute simulation timespan. The goal of this application was to evaluate multiple different network events at isolated simulation time intervals, which reduces packet collision or interference. Relative to an elapsed-time timeline starting at $t = 0$, the first event occurs at $t = 0$ seconds and consists of a UDP unicast of telemetry from the back of the satellite train (node 10.1.1.40) towards the front satellite in the train (node 10.1.1.70) on the 53 degree inclination plane. This event could be representative of situational awareness communications between intra-plane satellite train nodes. The second event occurs at $t =$ 150 seconds, and consists of a UDP multicast of telemetry from node 10.1.1.70 to the other nodes on the 53 degree inclination plane. This event could also be representative of situational awareness communication. In this case, the situational awareness could perhaps be an emergency broadcast from the leader node about future proximity operations or collision avoidance. The third event occurs at $t = 300$ seconds, and consists of a TCP unicast of download data to a relay node 10.1.1.30. The fourth event occurs at $t = 450$ seconds, and consists of a TCP unicast of download data to the destination node 10.1.1.10. Events three and four resemble a Delay-Tolerant Network (described in subsection 2.1.3). The config file for this scenario is located in the appendix (see A).

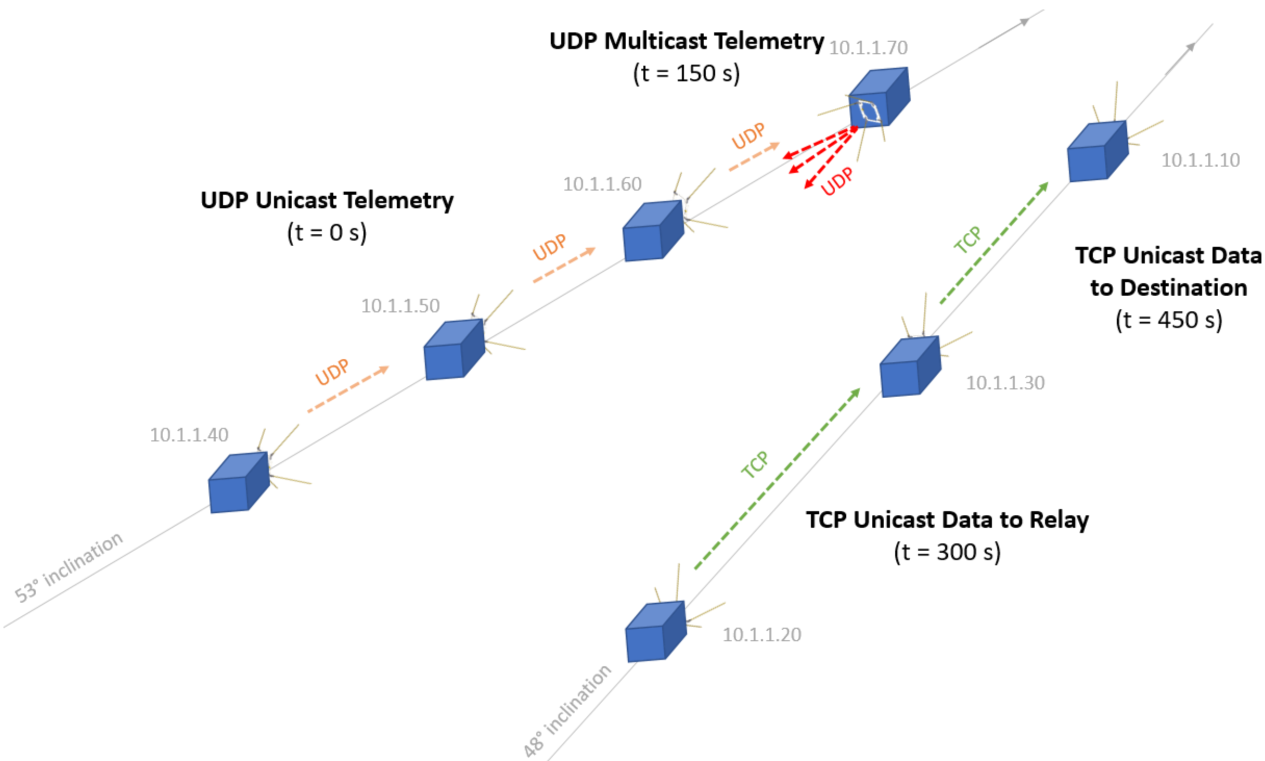

Figure 5.3: **Application 1 Network Events**. Four different network events occur in application 1: UDP unicast of telemetry, UDP multicast of telemetry, TCP unicast data transmission to a relay, and TCP unicast data to the destination node.

After the initial configuration file runs, the spacecraft state data is generated as part of the Astrodynamics module. Next, the Event Planning module generates the relative position and velocity data. In this case, the relative position and velocity statistics do not need to be generated, since the events in this application do not cross from one plane to the next, and the position of each node in the satellite train relative to the others remains fixed. Same relative position relative to node 10.1.1.70 is shown below in Figure 5.4.

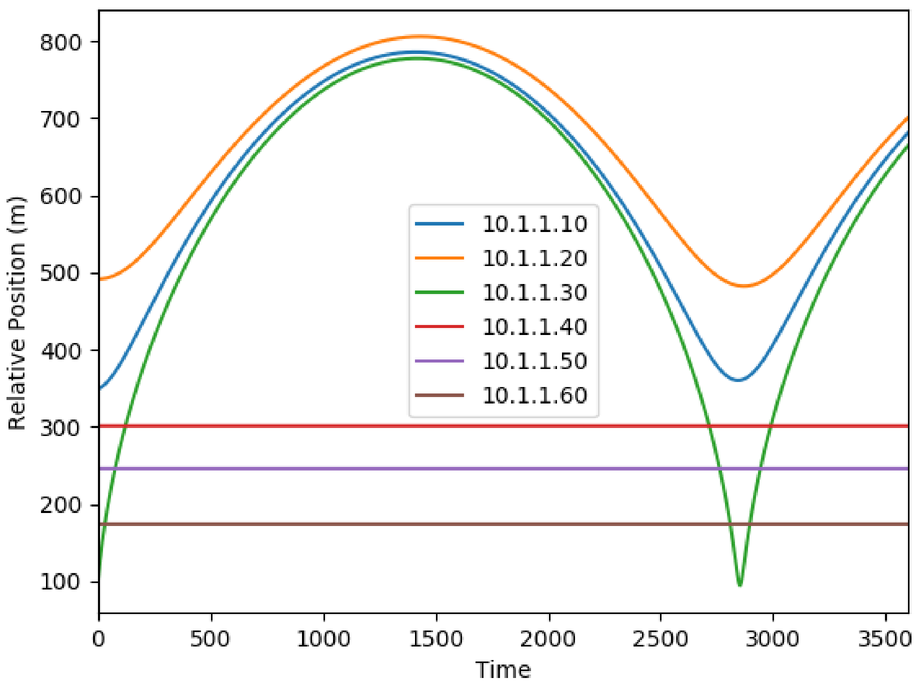

Figure 5.4: **Relative Position Plot for 10.1.1.70**. The relative position plot for 10.1.1.70 shows the constant separation from the other three nodes in the 53 degree inclination plane, and variable separation from the three nodes in the 48 degree inclination plane.

Next, the network configuration file is created and passed to ns-3 (see A). The Network Analysis module generates time series data plots for the network flows, and a summary table of the network statistics. There are several options for the time series plots—kilobytes transmitted per second, kilobytes lost per second, packets transmitted per second, and packets lost per second. Since ns-3 generates data as flows from transmitting nodes to receiving nodes, the traffic is categorized by source ID. Each subplot is titled with the source ID of the flow with the destination IDs in the legend of each subplot.

The plot in Figure 5.5 displays the time series data in kilobytes transmitted per second. Referencing back to Figure 5.3, we can see the subplot for 10.1.10 shows the TCP acknowledgment for the TCP data transmitted from 10.1.1.30 at 450 seconds. Note that although the plots for 10.1.1.10, 10.1.1.20, and 10.1.1.30 share the same shape, the TCP acknowledgment in 10.1.1.10 is 33 kbps whereas the actual TCP data transmission is near 800 kbps. The plot for 10.1.1.20 shows the TCP relay to 10.1.1.30 at 300 seconds and the smaller UDP telemetry transmission. The plot for 10.1.1.30 shows the TCP relay to 10.1.1.10, as well as the UDP telemetry and the TCP acknowledgment from node 10.1.1.20. Again, the TCP acknowledgment has a similar shape to transmision, but is much lower in magnitude. The plots for nodes 10.1.1.40, 10.1.1.50, and 10.1.1.60 all show UDP telemetry throughout the entire time period send to 10.1.1.70. Finally, the plot for 10.1.1.70 shows the UDP transmissions for the three other nodes in the 53 degree inclination plane. Each of these four plots share the same characteristics, which is due to the fact that the UDP transmission is sent in a repeated On-Off pattern at a constant level of 1.4 kbps for the duration of the transmission. The important note for these plots is that the unicast telemetry for nodes 10.1.1.40, 10.1.1.50, and 10.1.1.60 all start at $t = 0$, whereas the multicast telemetry for node 10.1.1.70 starts at $t = 150$ seconds.

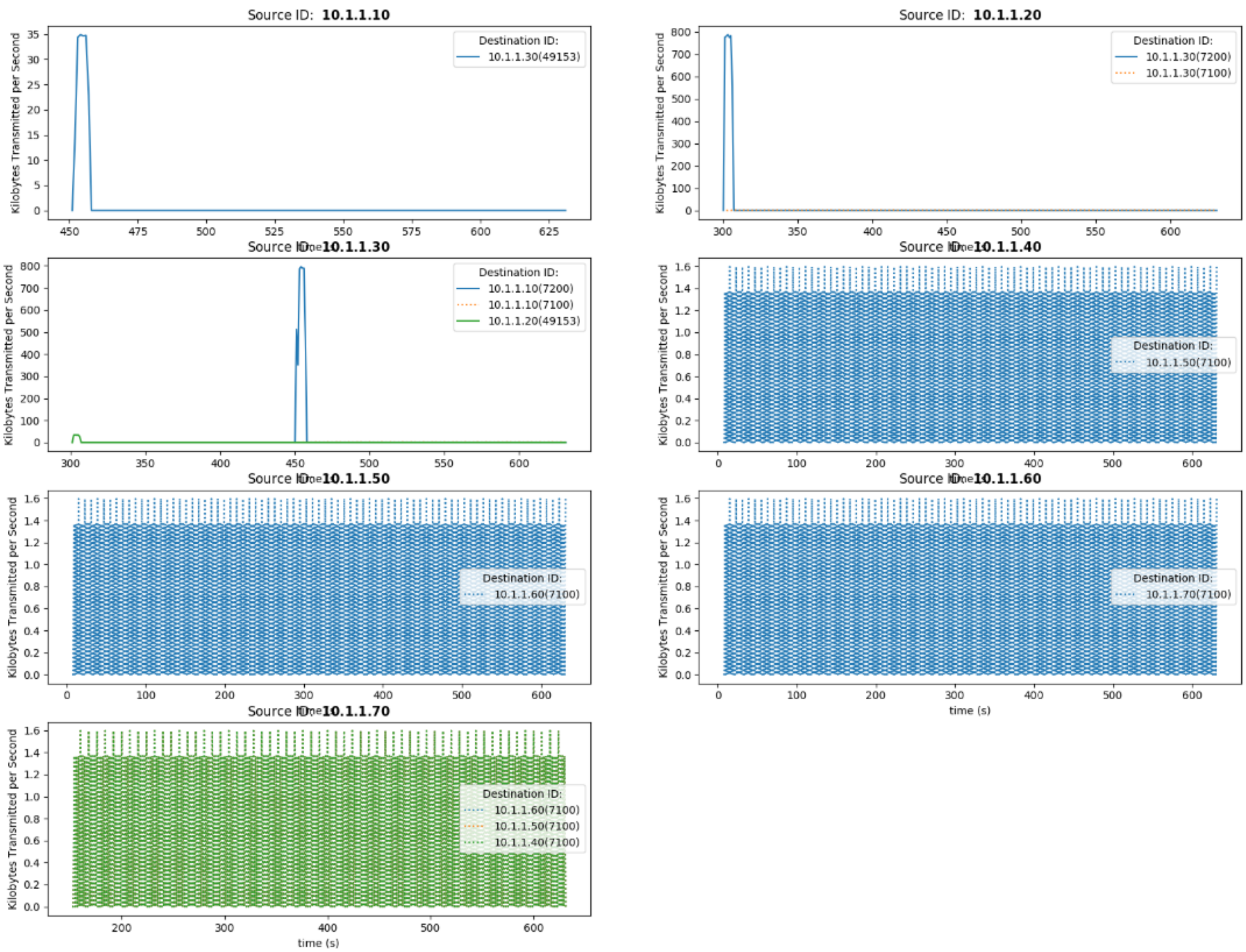

Figure 5.5: **Network Activity for Application 1: Kilobytes Transmitted Per Second**. Each subplot is titled with the source ID of the network flow. The destination IDs are given in the legends of each plot. The numbers in parenthesis refer to the type of communication: 7200 is a TCP transmission, 7100 is a UDP transmission, and 49153 is a TCP acknowledgment.

The plot in Figure 5.6 shows the time series data in kilobytes lost per second. The important takeaway from this plot is that the UDP transmissions remain fairly lossless, while the TCP transmissions do have some loss. Referring to the application diagram in Figure 5.3, it is evident that these TCP transmission losses occur at the beginning of each event, which indicates packets are dropped as nodes are turned on. The TCP acknowledgements are also fairly lossless. The UDP transmission from 10.1.1.40 to 10.1.1.50 is the only UDP transmission with loss, and this occurs at $t = 450$ seconds when the multicast event from 10.1.1.70 is turned on. A possible reason for this could be packet collision due to the larger relative distance between the two nodes, though it remains small as an instantaneous peak loss of 0.2 kbps.

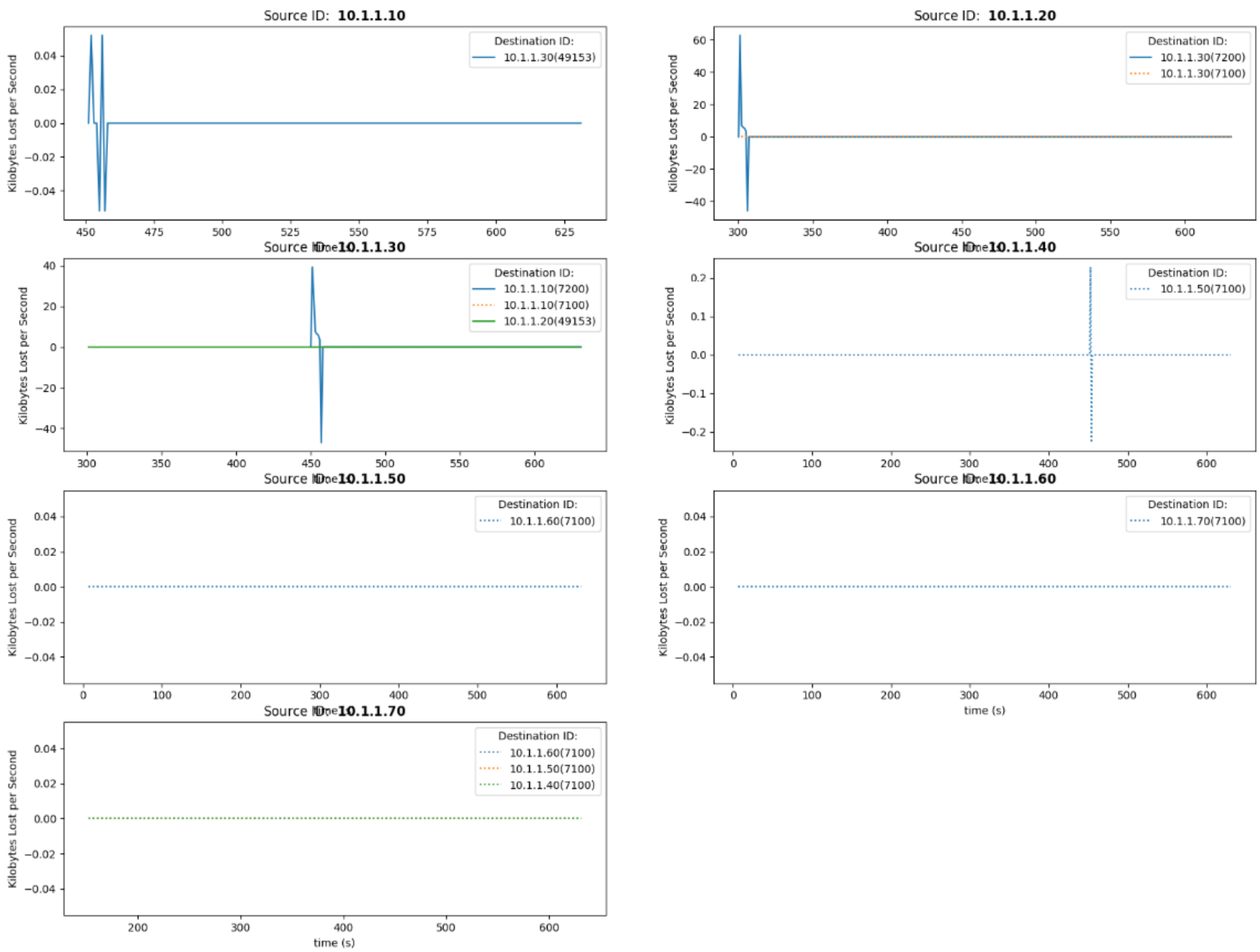

Figure 5.6: **Network Activity for Application 1: Kilobytes Lost Per Second**. Each subplot is titled with the source ID of the network flow. The destination IDs are given in the legends of each plot. The numbers in parenthesis refer to the type of communication: 7200 is a TCP transmission, 7100 is a UDP transmission, and 49153 is a TCP acknowledgment.

Table 5.1 shows the network statistics of this application. The data is summarized by flow, much like the time series plot above. Each flow is labelled by class, and several categories of network statistics. There are two "download" flows, which corresponds to the TCP relay transmissions. The jitter is 1.5 ms or less, which falls under the 30 ms Cisco quality of service recommendation. The latency is 66.5 ms or less, which falls under the 150 ms Cisco quality of service recommendation. The PER is less than 1% Cisco quality of service recommendation. An important note about the runtime is duration of the transmission, which is around 6-7 seconds. All of the other flows have much longer runtimes, since the UDP operates on a continual "OnOff" mode. Comparing the Rx Bytes and Tx Bytes columns indicates the total packet loss in each flow. Much of this packet loss occurs right at the beginning of events (at 0, 150, 300, and 450 seconds), which indicates dropped packets associated with the node being turned on and off.

Table 5.1: **Application 1 Network Statistics**.

Source	Destination	Class	Jitter (ms)	Latency (ms)	PER (%)	Runtime (s)	Rx Bytes	Tx Bytes	Throughput (bps)
10.1.1.20	10.1.1.30	download	1.5	64.5	0.869	5.756	4422340	4461148	6146039.643
10.1.1.30	10.1.1.10	download	1.5	66.5	0.876	6.655	4452576	4491972	5352207.593
10.1.1.20	10.1.1.30	telemetry	0.5	0.5	0	328.800	235068	235068	5719.399345
10.1.1.30	10.1.1.10	telemetry	0.5	0.5	0	178.760	128136	128136	5734.406426
10.1.1.40	10.1.1.50	telemetry	1.5	2.5	0	624.802	445968	445968	5710.191434
10.1.1.50	10.1.1.60	telemetry	1.5	2.5	0	624.801	445968	445968	5710.201343
10.1.1.60	10.1.1.70	telemetry	1.5	2.5	0	624.801	445968	445968	5710.204015
10.1.1.70	10.1.1.60	telemetry	0.5	0.5	0	478.680	341772	341772	5711.89629
10.1.1.70	10.1.1.50	telemetry	0.5	1.5	0	478.681	341772	341772	5711.887647
10.1.1.70	10.1.1.40	telemetry	0.5	1.5	0	478.681	341772	341772	5711.884157

This application is a good example of several types of wireless network activity that could be implemented on a dual-plane satellite train network. This application looked at UDP unicast relay, UDP multicast, and a TCP relay. These communications were evaluated along intra-plane satellite trains, which is a fairly reliable communication method since relative distances between nodes remain constant. This application included some analysis of network metrics and network flows, which have performed as expected. Further parameter studies would be beneficial in extending this analysis.

The implementation in Spacecraft-ns3 requires modification to two config files and a couple command-line arguments to run. This application has shown the type of plots and data that the Network Analysis module returns. The time series data is a good visualization of the network at any given time, and the network statistics summarizes the network performance with quality of service metrics.

5.3 Application 2: Cross-Plane TCP and UDP Exchange

The second application of the dual-plane satellite train case study evaluates four events in an eight minute simulation timespan. The goal of this application was to evaluate crosslinks between planes centered around the point at which the two planes cross. Finding this intersection of planes required the Event Planning Module (see Table 5.2). Since antennas are fixed to a spacecraft node with fixed attitude, it is likely more packets may be dropped in this application. Events are generated back-to-back, which could result in interesting network analysis at the event switch timestamps.

The timeline for this application was structured around the timestamp at which the two planes cross. The total scenario spans from May 1st, 2020 from 08:00:00 UTC to May 1st, 2020 from 09:00:00 UTC. The planes cross near 08:48:33 UTC, and at this point the nodes 10.1.1.30 and 10.1.1.70 have the smallest relative distance at 94.507 meters. This timestamp is given by the first row in Table 5.2, as the elapsed time $t = 2853$ seconds from the scenario start time at 08:00:00 UTC. Several nodes are closest to node 10.1.1.30 at the time index 0 (corresponding to 08:00:00 UTC), which is when another cross-plane intersection occurs.

Table 5.2: **Event Planning for Application 2**. This table shows the data resulting from the Event Planning module. This table is relative motion to 10.1.1.30. The time index is relative to the scenario start time.

Name	Time Index	Distance (m)
10.1.1.70	2853	94.506
10.1.1.60	0	117.692
10.1.1.50	0	246.795
10.1.1.40	0	301.375
10.1.1.10	2840	347.648
10.1.1.20	2871	491.636

The relative position and velocity plots visually display this information. From these plots, it is evident that the two planes cross around 2850 elapsed seconds. Knowing this, we know that at around 1450 elapsed seconds the two planes are at the widest point of the relative motion.

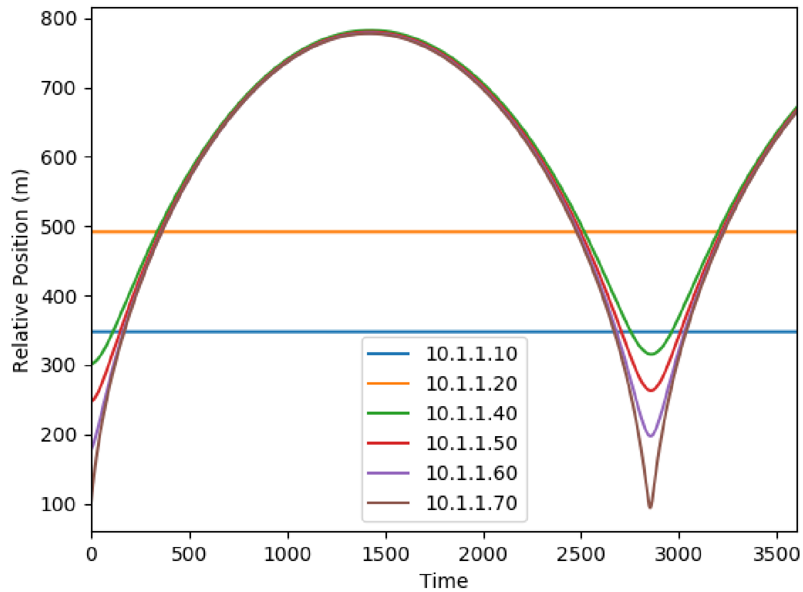

Figure 5.7: **Position Relative to Node 10.1.1.30.** This plot shows the relative position to Node 10.1.1.30, and the point of closest approach occurs around 2800 seconds.

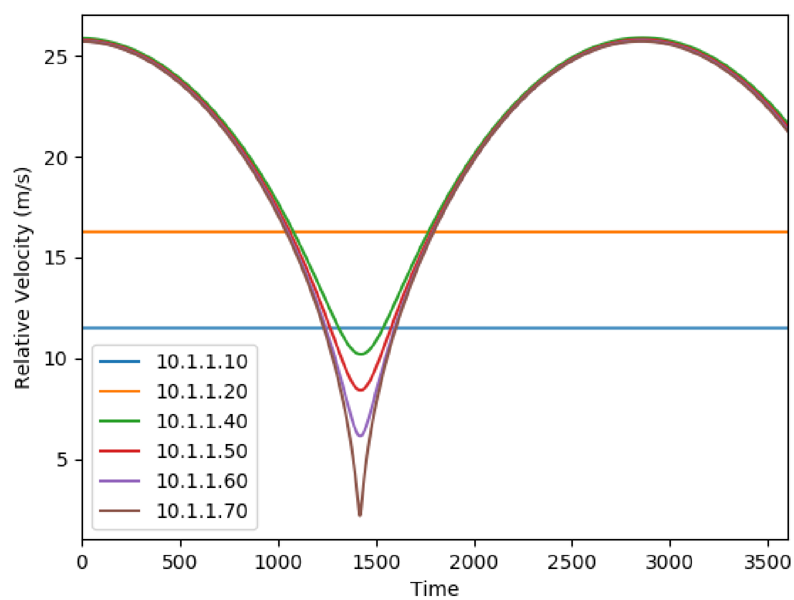

Figure 5.8: **Velocity Relative to Node 10.1.1.30.** This plot shows the relative velocity to Node 10.1.1.30, and the point of furthest approach occurs around 1400 seconds.

Relative to an elapsed-time timeline starting at $t = 0$, the first event occurs at $t = 0$ and consists of TCP data aggregation from the outer nodes on the 48 degree inclination plane (10.1.1.10 and 10.1.1.20) towards the center node (10.1.1.30). This event could be representative of a satellite train aggregating sensing data to be sent to another satellite train or cluster. The second event occurs at $t = 120$ seconds, and consists of a UDP telemetry and TCP data crosslink from the center node on the 48 degree plane (10.1.1.30) towards the front node on the 53 degree plane (10.1.1.70). This event is centered around the timestamp at which the the transmission and receiving node are closest, a timestep that is generated from the Event Planning module (see section 4.3). The third event occurs at $t = 240$ s, and consists of a TCP relay from the front node to the back node of the satellite train on the 53 inclination plane. The middle nodes (10.1.1.50 and 10.1.1.60) are relay nodes. This event is representative of a distribution of sensing data received from the cross-plane satellite train. The fourth event occurs at $t = 360$ seconds and consists of another crosslink, this time from the 53 inclination plane (10.1.1.40) to the 48 inclination plane (10.1.1.20). The config file for this scenario is located in the appendix (see B). This scenario uses high data volume transmissions, which was used to tax the network capacity and evaluate corresponding network behavior. The first event is a data aggregation of 1 Gb from the outer node to the center node, the second event is a crosslink of 2 Gb, the third event is a distribution of 2 Gb throughout the second satellite train, and the fourth event is a crosslink of 1 Gb to the first satellite train.

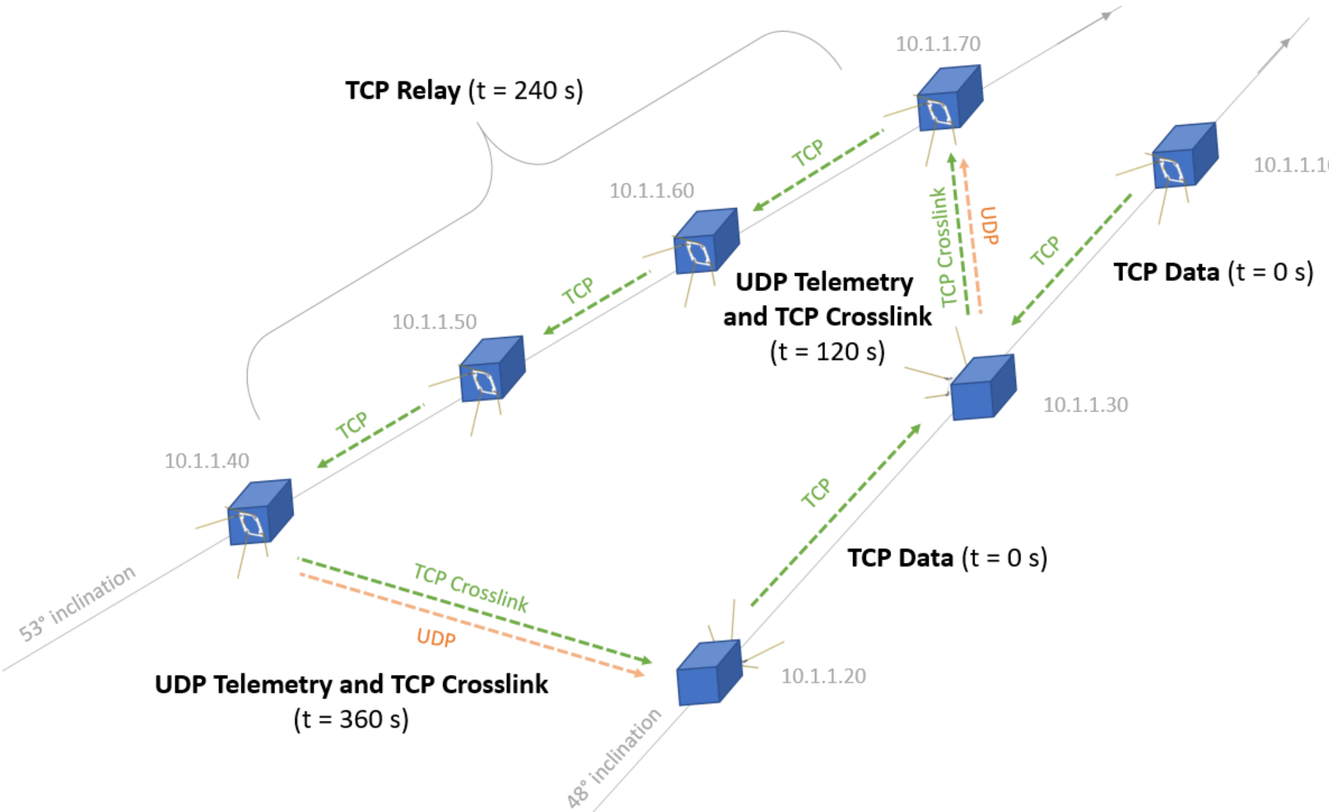

Figure 5.9: **Application 2 Network Events**. Four different network events occur in application 2: TCP unicast data aggregation, unicast TCP data and UDP telemetry planar crosslink, TCP data relay, and unicast TCP data and UDP telemetry planar crosslink.

Figure 5.10 displays the time series data of network activity in terms of kilobytes transmitted per second. It is immediately apparent that this plot looks much different than those of the first application. This application requires only five subplots rather than seven, since nodes 10.1.1.50 and 10.1.1.60 are passive relay nodes and do not generate any network traffic. The top two plots show the data aggregation from nodes 10.1.1.10 and 10.1.1.20 towards node 10.1.1.30 starting at $t = 0$ seconds. The task assigned to these two nodes is to each transmit 1 Gb of data to node 10.1.1.30. Upon further analysis of the data, node 10.1.1.10 transmitted 52,017,528 bytes out of 1 Gb (5.2%) by $t = 120$ seconds and node 10.1.1.20 transmitted 50,681,004 bytes out of 1 Gb (5.0%) by $t = 120$ seconds. These nodes were left “on” to to evaluate the remaining behavior. The subplot for node 10.1.1.30 shows the crosslink from node 10.1.1.30 to node 10.1.1.70. This transmission occurs in two separate segments, with a middle section of approximately 40 seconds in which it drops out. Additionally in this plot we can see the UDP telemetry transmission, and the TCP transmission and acknowledgment from node 10.1.1.20. The plot for node 10.1.1.40 shows the TCP relay from node 10.1.1.70 to node 10.1.1.40, which centers around 21 kbps for most of the transmission, and then increases to 35 kbps around $t = 420$ seconds. Additionally, we can see the TCP data transmission from 10.1.1.40 to 10.1.1.20 starting at $t = 360$ seconds. It is much lower since the antennas associated in this crosslink do not have direct line-of-sight. The beamwidth for these antennas is a cone extending 30 degrees from the beam axis that is fixed along the velocity direction, so the smaller transmission may have resulted from an indirect link. The subplot for 10.1.1.70 mirrors the shape in the 10.1.1.40 subplot, which is because the TCP relay is shown in the former plot and the TCP acknowledgement is shown in the latter plot.

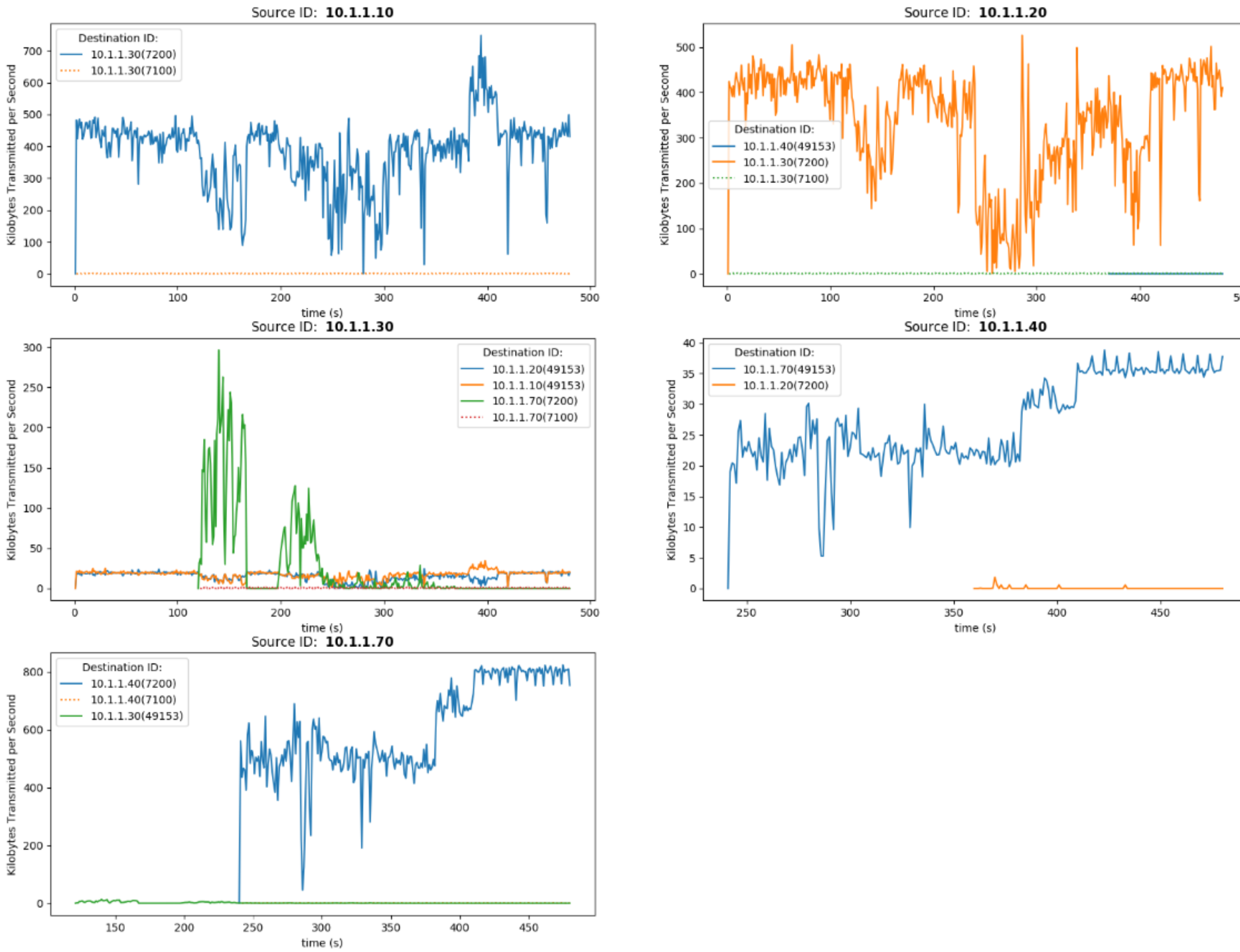

Figure 5.10: **Network Activity for Application 2: Kilobytes Transmitted Per Second**. Each subplot is titled with the source ID of the network flow. The destination IDs are given in the legends of each plot. The numbers in parenthesis refer to the type of communication: 7200 is a TCP transmission, 7100 is a UDP transmission, and 49153 is a TCP acknowledgment.

Figure 5.11 displays the time series data of network activity in terms of kilobytes lost per second. As with application 1, the TCP data has lossy behavior and the UDP telemetry transmissions have practically no loss. Overall, the top two plots for nodes 10.1.1.10 and 10.1.1.20 sustain more loss than the other transmissions, though the relay transmission from 10.1.1.70 to 10.1.1.40 has the highest instantaneous peak at $t = 240$ seconds during the start of the third event.

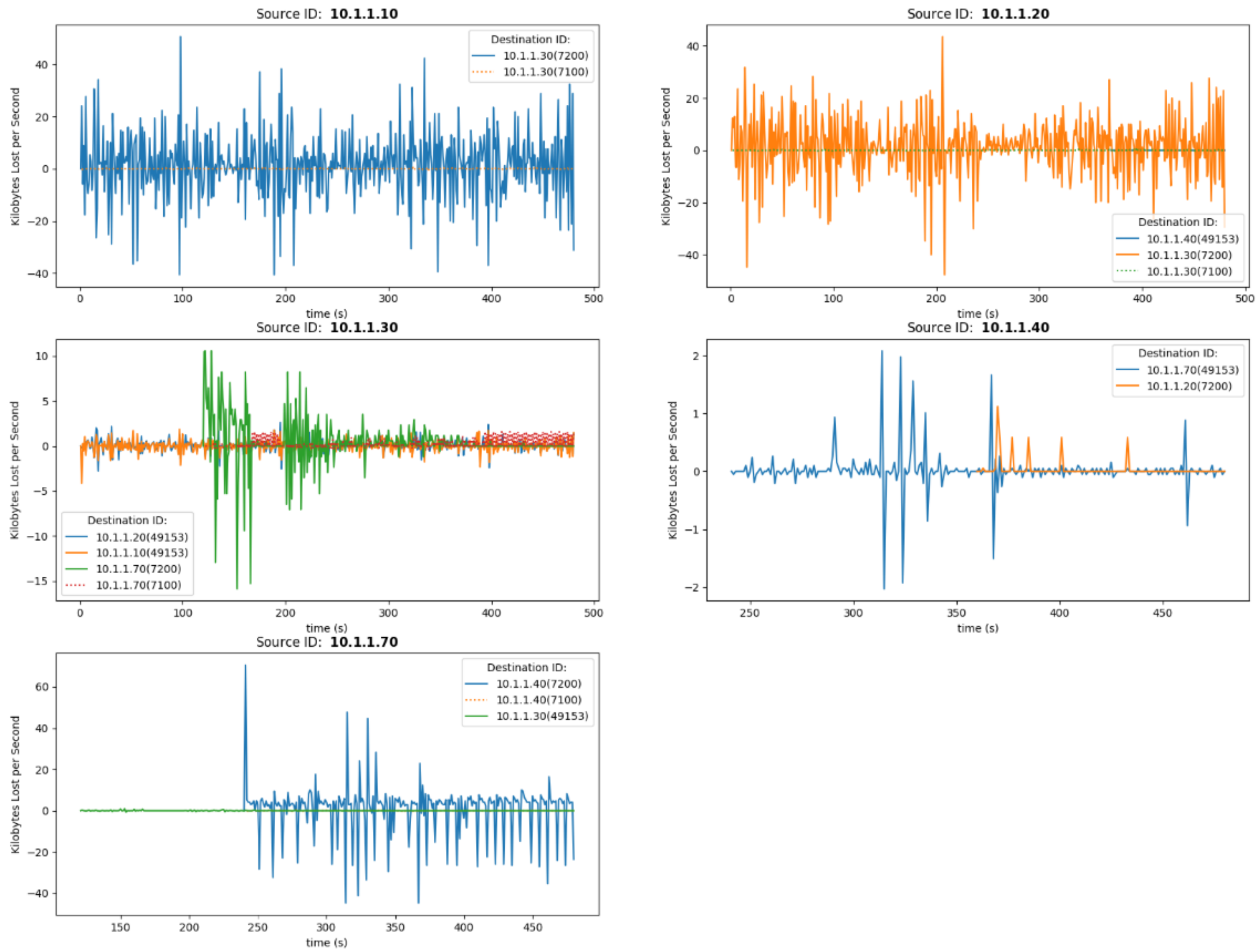

Figure 5.11: **Network Activity for Application 2: Kilobytes Lost Per Second**. Each subplot is titled with the source ID of the network flow. The destination IDs are given in the legends of each plot. The numbers in parenthesis refer to the type of communication: 7200 is a TCP transmission, 7100 is a UDP transmission, and 49153 is a TCP acknowledgment.

Table 5.3 shows the network statistics of Application 2. The data is summarized in flows characterized by source ID to destination ID. Most of the flows in this scenario are "download", which are the TCP data transmissions. The jitter rises above the Cisco quality of service recommendation of 30 ms in several cases, most notably in the telemetry transmission from 10.1.1.30 to 10.1.1.70. The jitter is higher for telemetry than download, which is generally a characteristic found in UDP. The latency is generally around 100 ms for all transmissions, and spikes up to nearly 200 ms in the same telemetry 10.1.1.30 to 10.1.1.70 transmission. Cisco recommends latencies of no more than 150 ms for one-way transmissions, which many of these fall under. The PER varies widely between applications, and is notably large for the telemetry transmission from 10.1.1.30 to 10.1.1.70 and the download from 10.1.1.40 to 10.1.1.20. The first case is likely due to the jitter and latency already described, and the second case is likely due to the lack of a direct beam-axis antenna to beam-axis antenna link.

Table 5.3: **Application 2 Network Statistics**.

Source	Destination	Class	Jitter (ms)	Latency (ms)	PER (%)	Runtime (s)	Rx Bytes	Tx Bytes	Throughput (bps)
10.1.1.10	10.1.1.30	download	5.5	99.5	0.38	480.00	183465516	184194048	3057763.357
10.1.1.10	10.1.1.30	telemetry	64.5	103.5	0.53	478.76	339720	341772	5676.657891
10.1.1.20	10.1.1.30	download	5.5	97.5	0.36	480.00	160449432	161066832	2674159.012
10.1.1.20	10.1.1.30	telemetry	73.5	110.5	1.07	478.70	337896	341772	5646.911529
10.1.1.30	10.1.1.70	telemetry	152.5	194.5	38.79	270.77	148428	256272	4385.349748
10.1.1.30	10.1.1.70	download	11.5	128.5	1.52	247.65	9618024	9766788	310703.2103
10.1.1.70	10.1.1.40	telemetry	55.5	111.5	0.53	238.72	169860	170772	5692.415181
10.1.1.70	10.1.1.40	download	3.5	106.5	0.12	240.00	144263340	144469728	4808792.775
10.1.1.40	10.1.1.20	download	14.5	87.5	69.23	12.54	1284	5512	819.0581874

This application represents several more types of wireless network activity that could be implemented on a dual-plane satellite train network. This application focuses on a more data-heavy approach with closely clustered events around a significant timestamp generated by the Event Planning module. The resulting network behavior is more lossy, as expected.

The data aggregation in the first event took much longer than expected, which likely affected the resulting network events. Since the first event did not fully transmit in the given time interval, the following events would not have been able to fully transmit the same data. Future studies into the correlation between antenna power, transmission time, and data size would be necessary to optimize network performance.

5.4 Discussion of Applications

This chapter evaluated two applications of a dual-plane satellite train network case study modeled after the Starlink Block V1.0 satellites. The first application evaluated inter-plane satellite train communications, including UDP unicast relay, UDP multicast, and low-volume TCP data relay transmitted in a “store-and-forward” method. Network events were spaced across a ten minute window to avoid interference from competing events. This application did not tax the network, and it is evident that all network activities were completed within CISCO quality of service recommendations. The second application was much more data-heavy, and consisted of TCP data aggregation, two TCP and UDP inter-plane crosslinks, and TCP data relay across a 4-node satellite train. This application was much more taxing on the network, and none of the events transmitted fully within the allotted time intervals. There were some network flows that failed the CISCO quality of service recommendations.

These applications show the analytical capabilities of the Spacecraft-ns3 simulator in rigorously quantifying network performance over a range of metrics. To extend this research further, parameter studies would be necessary to evaluate additional applications and optimize network performance. Some of these future work recommendations are given in chapter 6.

Chapter 6

Conclusions

The research scope of this thesis was to extend current space network simulation efforts towards space-based networks primarily comprising of inter-satellite links using RF and IP-based networking. As part of this research, a literature review was conducted to establish the current state of space networking research. Fundamental topics were presented to give context to the tool developed in this research, Spacecraft-ns3. Spacecraft-ns3 was then applied towards an exploratory next-generation space network case study.

Two research questions were considered as part of the research scope. Each question is answered below:

Question 1: How can the gap between network simulation and astrodynamics simulation be effectively and adequately bridged, without taxing computational resources or suffering from low-fidelity model assumptions?

Many approaches have been considered to integrate network simulation with astrodynamics simulation. A common approach that has been used by NASA is to integrate the astrodynamics simulator STK with the network simulator QualNet. This approach is effective, but taxes computational resources due to the bulky nature of these high-fidelity simulators. Other approaches consisting of custom models additions to network simulators are often low-fidelity due to their application-specific usage. Spacecraft-ns3 bridges this gap by providing high fidelity astrodynamics models as an extension to the network simulator ns-3.

Spacecraft-ns3 further extends the functionality of an integrated space network simulation tool by providing a discrete-event planning module and a network analysis module.

Question 2: What do expected future space networks look like, and how can this tool provide users with adequate modeling, simulation, and data analysis of these networks?

Next-generation space networks are increasingly distributed, space-based, and rely on autonomy and inter-satellite communications. Current research into space networking focuses on the Internet of Space Things paradigm, commercial megaconstellations, and space-based 4G LTE and 5G networks. The Spacecraft-ns3 tool is application-generic, and can support both TLE and Keplerian orbital element definitions. Spacecraft-ns3 supports platform and antenna attitude determination. In terms of network applications, Spacecraft-ns3 supports all 802.11 networks and provides users with flow-based network performance statistics. Several visualization tools are included in Spacecraft-ns3, which are used to analyze relative spacecraft motion and network activity. Spacecraft-ns3 was then used in a dual-plane satellite train configuration to analyze network activity in a next-generation satellite network.

6.1 Summary

Chapter One gave an introduction to the topic of space network simulation and presented motivation for this research. It argues that space network simulation involves both networking and astrodynamics simulation, and requires a flexible and high-fidelity integrated simulation design.

With this goal in mind, a literature review of prior space networking research was conducted in Chapter Two, which included a discussion of space networks and evolving space architectures, implementation of IP technologies in the space regime, and current research into space-based networking. A survey of prior space network simulation tools was surveyed, which provided additional motivation for the work. Although there have been several tools designed for space networking research, no existing tool effectively implements an astrodynamics simulator that is flexible and high-fidelity.

Chapter Three provided an overview of fundamental concepts necessary for space networking simulation. This chapter was divided into three parts—orbital mechanics concepts used in space networking, networking concepts used for space applications, and simulation of space networks. This chapter was necessary to provide general background knowledge for Chapter 4 and Chapter 5.

Chapter Four introduced Spacecraft-ns3, an integrated simulator designed as an extension to the ns-3 simulator. This extension comprises of three module—the astrodynamics module, the event planning module, and the space network analysis module. The Astrodynamics module propagates orbits of spacecraft nodes and defines spacecraft attitude time history. The Event Planning module addresses the need for mechanisms to predict opportune times for network activity. The Network Analysis module configures the network simulator and analyzes the output.

Chapter Five introduced a case study using Spacecraft-ns3. The motivation for this case study comes from literature reviews done in chapter 2. A modified version of a subset of the Starlink megaconstellation is evaluated to extend current on-orbit spacecraft constellations with space networking capabilities. This chapter follows two applications of the case study in Spacecraft-ns3 and discusses the resulting network data.

Chapter Six summarizes this research and provides future work recommendations.

6.2 Future Work

This research presents a simulator tool used for space network analysis. This work has focused mainly on exploratory network analysis rather than detailed parameter studies. To extend this work further, there are several different avenues of research that could be expanded upon:

- **Expansion of Astrodynamics.** There are currently only three orbit-frame attitude profiles that are supported in Spacecraft-ns3. An expansion on the attitude determination modeling would be to extend the coordinate frame analysis to define spacecraft body frames uniquely from the orbit frames. Furthermore, currently antennas are fixed to the spacecraft body frame and are defined in the network configuration YAML file. This only allows for static representation of the antennas. It would be beneficial to enable dynamic antenna pointing, so that antenna pointing could be defined by both static and dynamic methods. Dynamic pointing would require the use of antenna pointing data files, which would likely use relative position data.

- **Expansion of Event Planning.** The event planning methods used in this research center on analysis of relative position of cross-plane intersections. This could be expanded to more ad-hoc methods in which timestamps and time intervals are identified in which network nodes fall under a maximum relative distance parameter.

- **Network Parameter Studies.** It would be beneficial to analyze link budgets in the literature as a basis for developing a link budget module. Libraries of standardized platforms and components would be useful in developing more realistic network events and activity.

- **Automation of Spacecraft and Network YAML Files.** The current implementation of Spacecraft-ns3 uses user-defined spacecraft and network YAML configuration files. It would be beneficial to remove some of the manual work necessary to configure network events. The user would then only need to configure the source node, destination node, and the standard network activity profile. Some examples of these profiles could be: "standard video stream", "high resolution imagery", or "stationkeeping data".

Bibliography

[1] P. Nicopolitidis, M. S. Obaidat, G. I. Papadimitriou, and A. S. Pomportsis, *Wireless networks*. Wiley Online Library, 2003.

[2] R. A. Slywezak, "Low-earth-orbit satellite internet protocol communications concept and design," *NASA Glenn Research Center*, 2004.

[3] I. F. Akyildiz and X. Wang. Wireless mesh networks. [Online]. Available: https://slideplayer.com/slide/6985455/

[4] B. J. Barritt and W. M. Eddy, "Astrolink for modeling, simulation, and operation of aerospace communication networks," in *32nd AIAA International Communications Satellite Systems Conference*, 2014, p. 4438.

[5] J. L. Barnes, G. J. Clark, and W. Eddy, "Cogswel: A network emulator for cognitive space networks," in *34th AIAA International Communications Satellite Systems Conference*, 2016, p. 5763.

[6] A. J. Turner, "An open-source, extensible spacecraft simulation and modeling environment framework," Ph.D. dissertation, Virginia Tech, 2003.

[7] J. A. Ruiz-de Azúa, C. Aragur, A. Calveras, E. Alarcón, and A. Camps, "Towards an integral model-based simulator for autonomous earth observation satellite networks," in *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2018, pp. 7403–7406.

[8] C. Araguz, J. Ruiz de Azua Ortega, A. Calveras, A. Camps, and E. Alarcon, "Simulat-

ing distributed small satellite networks: A model-based tool tailored to decentralized resource-constrained systems," in *70th International Astronautical Congress*, 2019.

[9] B. Howard. (2019) What is agile aerospace? learn planet's approach. [Online]. Available: https://www.planet.com/pulse/what-is-agile-aerospace-learn-planets-approach/

[10] N. J. Boll, "Small satellites: Applications in resiliency and redundancy," in *AIAA SPACE 2016*, 2016, p. 5304.

[11] G. Namta. (2019-02-26) Let's talk about newspace. [Online]. Available: https://blog.satsearch.co/2019-02-26-lets-talk-about-newspace.html

[12] N. Goddard. (2000) Nasa turns on first internet node in space. [Online]. Available: https://www.spacedaily.com/news/internet-00l.html

[13] W. Ivancic, D. Stewart, D. Shell, L. Wood, P. Paulsen, C. Jackson, D. Hodgson, J. Notham, N. Bean, and E. Miller, "Secure, network-centric operations of a space-based asset: Cisco router in low earth orbit (cleo) and virtual mission operations center (vmoc)," *NASA Glenn Research Center*, 2005.

[14] K. Chamberlain. Internet in space: Bringing wifi to earth orbit, the moon, and mars. [Online]. Available: https://broadbandnow.com/report/internet-wifi-service-in-space/

[15] M. Benamar, S. Ahnana, F. Z. Saiyari, N. Benamar, M. D. El Ouadghiri, and J.-M. Bonnin, "Study of vdtn routing protocols performances in sparse and dense traffic in the presence of relay nodes." *J. Mobile Multimedia*, vol. 10, no. 1&2, pp. 78–93, 2014.

[16] D. Guanghong and X. Yi, "Tc-dtn: A tcp compatible dtn overlay protocol," in *2016 IEEE International Conference of Online Analysis and Computing Science (ICOACS)*. IEEE, 2016, pp. 320–326.

[17] D. E. Brooks, W. Eddy, S. K. Johnson, and G. J. Clark, “In-space networking on nasa’s scan testbed,” in *34th AIAA International Communications Satellite Systems Conference*, 2016, p. 5754.

[18] D. K. Kilcoyne, S. A. Rowe, W. C. Headley, D. J. Mortensen, R. W. McGwier, Z. J. Leffke, and R. C. Reinhart, “Link adaptation for mitigating earth-to-space propagation effects on the nasa scan testbed,” in *2016 IEEE Aerospace Conference*. IEEE, 2016, pp. 1–9.

[19] M. Rice, “Space-based networking using cubesat technologies,” in *33rd AIAA International Communications Satellite Systems Conference and Exhibition*, 2015, p. 4317.

[20] R. Alena, Y. Nakamura, N. Faber, and D. Mauro, “Heterogeneous spacecraft networks: Wireless network technology assessment,” in *2014 IEEE Aerospace Conference*. IEEE, 2014, pp. 1–13.

[21] M. Atiquzzaman and W. Ivancic, “Sigma for space sensor web networks,” in *ESTO AIST sensor web technology meeting*, 2007, pp. 2–3.

[22] I. F. Akyildiz and A. Kak, “The internet of space things/cubesats: A ubiquitous cyber-physical system for the connected world,” *Computer Networks*, vol. 150, pp. 134–149, 2019.

[23] Mjjsnell. (2019-08-12) Interplanetary networking special integerest group: Dtn and the iot. [Online]. Available: http://ipnsig.org/

[24] T. Malik. The moon will get its own mobile phone network in 2019. [Online]. Available: https://www.space.com/39835-moon-mobile-phone-network-ptscientists-2019.html

[25] H. J. Kramer. (2020) Starlink. [Online]. Available: https://directory.eoportal.org/web/eoportal/satellite-missions/s/starlink

[26] A. Mann. (2020-01-17) Starlink: Spacex's satellite internet project. [Online]. Available: https://www.space.com/spacex-starlink-satellites.html

[27] S. Clark. (2020-03-30) Oneweb files for bankruptcy. [Online]. Available: https://spaceflightnow.com/2020/03/30/oneweb-files-for-bankruptcy/

[28] K. Bhasin and J. L. Hayden, "Space internet architectures and technologies for nasa enterprises," in *2001 IEEE Aerospace Conference Proceedings (Cat. No. 01TH8542)*, vol. 2. IEEE, 2001, pp. 2–931.

[29] R. Radhakrishnan, W. W. Edmonson, F. Afghah, R. M. Rodriguez-Osorio, F. Pinto, and S. C. Burleigh, "Survey of inter-satellite communication for small satellite systems: Physical layer to network layer view," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 4, pp. 2442–2473, 2016.

[30] R. Welle and T. Rose. (2017-07-12) Small-satellite laser communications. [Online]. Available: https://www.nasa.gov/sites/default/files/atoms/files/07_aerospace_corp_small_satellite_laser_communications_0.pdf

[31] B. Yost. (2019-03-12) State of the art of small spacecraft technology: Communications. [Online]. Available: https://sst-soa.arc.nasa.gov/09-communications

[32] B. Barritt, K. Bhasin, W. Eddy, and S. Matthews, "Unified approach to modeling & simulation of space communication networks and systems," in *2010 IEEE International Systems Conference*. IEEE, 2010, pp. 133–136.

[33] E. H. Jennings, J. S. Segui, and S. Woo, "Machete: environment for space networking evaluation," *Pasadena, CA: Jet Propulsion Laboratory, National Aeronautics and Space ...*, 2010.

[34] E. Jennings, R. Borgen, S. Nguyen, J. Segui, T. Stoenescu, S.-Y. Wang, S. Woo, B. Barritt, C. Chevalier, and W. Eddy, "Space communications and navigation (scan) network simulation tool development and its use cases," in *AIAA Modeling and Simulation Technologies Conference*, 2009, p. 6036.

[35] A.-J. Merts and A. Barnard, "Simulating manets: A study using satellites with aodv and anthocnet," in *2016 Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech)*. IEEE, 2016, pp. 1–5.

[36] J.-H. Le Roux, "Development of a satellite network simulator tool and simulation of ax. 25, fx. 25 and a hybrid protocol for nano-satellite communications," Ph.D. dissertation, Stellenbosch: Stellenbosch University, 2014.

[37] J. A. Fraire, P. A. Ferreyra, and R. Velazco, "Glorbit: A 3d satellite orbit propagator for network topology analysis," *Revista de la Facultad de Ciencias Exactas, Físicas y Naturales*, vol. 3, no. 2, pp. 29–35, 2016.

[38] J. Puttonen, S. Rantanen, F. Laakso, J. Kurjenniemi, K. Aho, and G. Acar, "Satellite model for network simulator 3," *7th International ICST Conference on Simulation Tools and Techniques*, 2014.

[39] ns3. (2020) ns-3 ns3.14 documentation: Mobility. [Online]. Available: https://www.nsnam.org/docs/release/3.14/models/html/mobility.html

[40] B. Rhodes, "Skyfield: High precision research-grade positions for planets and Earth satellites generator," p. ascl:1907.024, Jul. 2019.

[41] smit009. (2015-07-12) The different frames and the keplerian elements. [Online]. Available: https://adcsforbeginners.wordpress.com/tag/earth-centred-inertial-frame/

[42] I. Analytical Graphics. (2016-09) Access tool. [Online]. Available: https://help.agi.com/stk/11.0.1/Content/stk/tools-01.htm

[43] F. R. Hoots, R. L. Roehrich, and T. Kelso, "Spacetrack report no. 3," *Colorado Springs CO: Air Force Aerospace Defence Command*, pp. 1–3, 1980.

[44] D. Vuckovic, P. Rajkovic, and D. Jankovic, "Guidelines for satellite tracking (navstar software)," in *2005 IEEE Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*. IEEE, 2005, pp. 713–716.

[45] J. Diebel, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," *Matrix*, vol. 58, no. 15-16, pp. 1–35, 2006.

[46] S. W. Shepperd, "Quaternion from rotation matrix," *Journal of Guidance and Control*, vol. 1, no. 3, pp. 223–224, 1978.

[47] F. L. Markley, "Unit quaternion from rotation matrix," *Journal of guidance, control, and dynamics*, vol. 31, no. 2, pp. 440–442, 2008.

[48] C. A. Balanis, *Antenna theory: analysis and design*. John wiley & sons, 2016.

[49] ns 3-dev. (2020) Design documentation - antennamodel. [Online]. Available: https://www.nsnam.org/docs/models/html/antenna-design.html#overview

[50] J. Banks, *Discrete event system simulation*. Pearson Education India, 2005.

[51] N. Hütter. (2018) Osi model. [Online]. Available: https://twitter.com/klavierler/status/1029366461851230208

[52] K. L. Calvert. (2006) Overview of ieee 802 standards. [Online]. Available: http://protocols.netlab.uky.edu/~calvert/classes/571/lectureslides/802overview.pdf

[53] R. Flickenger, *Wireless Networking in the Developing World: A practical guide to planning and building low-cost telecommunications infrastructure.* Hacker Friendly LLC, Seattle, WA, US, 2007.

[54] F. Farahmand. (2011) Sonoma state university lecture: Ieee 802.11 wireless lan standard. [Online]. Available: https://web.sonoma.edu/users/f/farahman/sonoma/courses/cet543/lectures/2011_Lectures/802WirelessLan.pdf

[55] M. Marszalek, M. Rummelhagen, and F. Schramm, "Potentials and limitations of ieee 802.11 for satellite swarms," in *2014 IEEE Aerospace Conference.* IEEE, 2014, pp. 1–9.

[56] L. Haury, "Applications of recent wireless standards in satellite networking," *Survey Paper Department of Computer Science Washington University in St. Louis*, 2008.

[57] N. Herscovici, C. Christodoulou, V. Lappas, G. Prassinos, A. Baker, and R. Magnuss, "Wireless sensor motes for small satellite applications," *IEEE Antennas and Propagation Magazine*, vol. 48, no. 5, pp. 175–179, 2006.

[58] ns 3. (2020) Doxygen ns-3 - the list of all attributes. [Online]. Available: https://www.nsnam.org/docs/release/3.6/doxygen/group____attribute_list.html

[59] J. A. R. de Azúa, A. Calveras, and A. Camps, "Internet of satellites (iosat): Analysis of network models and routing protocol requirements," *IEEE access*, vol. 6, pp. 20 390–20 411, 2018.

[60] G. Carneiro, P. Fortuna, and M. Ricardo, "Flowmonitor: a network monitoring framework for the network simulator 3 (ns-3)," in *Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools.* ICST (Institute for Computer Sciences, Social-Informatics and …, 2009, p. 1.

[61] T. Szigeti and C. Hattingh, "Quality of service design overview," *Cisco, San Jose, CA, Dec*, pp. 1–34, 2004.

[62] W. Mansouri, K. B. Ali, F. Zarai, and M. S. Obaidat, "Radio resource management for heterogeneous wireless networks: Schemes and simulation analysis," in *Modeling and Simulation of Computer Networks and Systems*. Elsevier, 2015, pp. 767–792.

[63] S. Gao, M. Brenchley, M. Unwin, C. I. Underwood, K. Clark, K. Maynard, L. Boland, and M. Sweeting, "Antennas for small satellites," in *2008 Loughborough Antennas and Propagation Conference*. IEEE, 2008, pp. 66–69.

[64] H.-N. Dai and Q. Zhao, "On the delay reduction of wireless ad hoc networks with directional antennas," *EURASIP Journal on Wireless Communications and Networking*, vol. 2015, no. 1, p. 16, 2015.

[65] T. Truckle. (2008-06-20) Radiation pattern of an antenna with main lobe in red, and back lobe in blue. [Online]. Available: https://en.wikipedia.org/wiki/Front-to-back_ratio#/media/File:Sidelobes_en.svg

[66] A. Chinmayi, M. Vasanthi, and T. R. Rao, "Performance evaluation of rf based inter satellite communication link at 60 ghz," in *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*. IEEE, 2016, pp. 59–64.

[67] ns 3-dev. (2020) ns3::friispropagationlossmodel class reference. [Online]. Available: https://www.nsnam.org/doxygen/classns3_1_1_friis_propagation_loss_model.html

[68] P. Kuruba and A. V. Sutagundar, "Space based wireless sensor network: A survey," *Special Issue: Advances in Cloud and Internet of Things*, 2017.

[69] O. Ullrich and D. Lückerath, “An introduction to discrete-event modeling and simulation,” *Simulation Notes Europe*, vol. 27, no. 1, pp. 9–16, 2017.

[70] B. Liang, “A discrete-event simulation tool for resource constrained networks,” Ph.D. dissertation, Citeseer, 2015.

[71] B. Rhodes. (2019) Skyfield: High precision research-grade positions for planets and earth satellites generator. [Online]. Available: https://pypi.org/project/skyfield/

[72] D. Vallado and P. Crawford, “Sgp4 orbit determination,” in *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, 2008, p. 6770.

[73] F. Bai and A. Helmy, “A survey of mobility models in wireless adhoc networks,” 2004.

[74] A. Mcgregor and D. Smithies, “Rate adaptation for 802.11 wireless networks: Minstrel,” *Submitted to ACM SIGCOMM*, 2010.

[75] D. Mohney. (2019) First batch of spacex starlink satellites in orbit. [Online]. Available: https://www.spaceitbridge.com/first-batch-of-spacex-starlink-satellites-in-orbit.htm

Appendices

Appendix A

Additional Information for Application 1

This section shows the full network config file used in Application 1.

```
# Application 1: TCP Relay and UDP Telemetry

VERSION: 0.1

config:
  startTime:          2673
  runTime:            480
  seed:               1     # random seed for simulation; cannot be zero


# configuration parameters for output tracing
tracing:
  prefix:             ./
  animFileName:       sns3-sim.xml
  traceFileName:      sns3-sim.tr
  routeFileName:      sns3-sim.routes
  neighFileName:      sns3-sim.neighbors
  flowFileBase:       flow_stats
  linkFileBase:       link_stats
  logInterval:        1.0
  verboseNet:         false
  verbose:            false
  specAnalyzerNode:   null

network:
  addressbase:        10.1.1.0
  bcastaddress:       10.1.1.255
  UDPPort:            7100
  TCPPort:            7200
  routing:            olsr
  routingAttributes:
      HelloInterval:  0.1

radio:
  standard:           802.11a
  staManager:         ns3::MinstrelWifiManager
  staManagerAttributes: null
  mac:
      Slot:                         36us   # 18us    # slot time
      Sifs:                         30us   # 20us    # SIFS duration
      AckTimeout:                   264us  # 176us   # ACK timeout duration
      CtsTimeout:                   264us  # 176us   # CTS timeout duration
      Rifs:                         6us    # 4us     # RIFS duration
      BasicBlockAckTimeout:         6us    # 4us     # Basic Block ACK timeout duration
      CompressedBlockAckTimeout:    336us  # 224us # Compressed Block ACK timeout duration
  dataMode:           OfdmRate6Mbps
  txPwrDbm:           20
  txPwrDbmMax:        30
  txPwrDbmMin:        0
  txPwrLevels:        31
  frequencyMHz:       2412
# ccaThreshold:       -50
```

Figure A.1: **Network configuration file for Application 1 (Part 1 of 2).**

```yaml
events:
  - {type: wifiRadioOn, address: 10.1.1.40, start: 5}
  - {type: wifiRadioOn, address: 10.1.1.50, start: 5}
  - {type: wifiRadioOn, address: 10.1.1.60, start: 5}
  - {type: wifiRadioOn, address: 10.1.1.70, start: 150}
  - {type: wifiRadioOn, address: 10.1.1.20, start: 300}
  - {type: wifiRadioOn, address: 10.1.1.30, start: 450}


nodes:

  - csvfile: case1/sat_48_b.csv
    address: 10.1.1.10
    antenna: {type: bicosine-backlobe, beamwidth-az: 30, beamwidth-el: 30, offset-az: 0, offset-el: 0}

  - csvfile: case1/sat_48_c.csv
    address: 10.1.1.20
    antenna: {type: bicosine-backlobe, beamwidth-az: 30, beamwidth-el: 30, offset-az: 0, offset-el: 0}
    applications:
      - {destAddress: 10.1.1.30, class: telemetry, start: 300, size:  200, type: OnOff, protocol: udp, rate: 10kbps}
      - {destAddress: 10.1.1.30, class: download,  start: 300, size:  400, type: BulkSend, bytes: 4e6}

  - csvfile: case1/sat_48_a.csv
    address: 10.1.1.30
    antenna: {type: bicosine-backlobe, beamwidth-az: 30, beamwidth-el: 30, offset-az: 0, offset-el: 0}
    applications:
      - {destAddress: 10.1.1.10, class: telemetry, start: 450, size:  200, type: OnOff, protocol: udp, rate: 10kbps}
      - {destAddress: 10.1.1.10, class: download,  start: 450, size:  400, type: BulkSend, bytes: 4e6}

  - csvfile: case1/sat_53_a.csv
    address: 10.1.1.40
    antenna: {type: omni}
    applications:
      - {destAddress: 10.1.1.50, class: telemetry, start: 5, size:  200, type: OnOff, protocol: udp, rate: 10kbps}

  - csvfile: case1/sat_53_b.csv
    address: 10.1.1.50
    antenna: {type: omni}
    applications:
      - {destAddress: 10.1.1.60, class: telemetry, start: 5, size:  200, type: OnOff, protocol: udp, rate: 10kbps}

  - csvfile: case1/sat_53_c.csv
    address: 10.1.1.60
    antenna: {type: omni}
    applications:
      - {destAddress: 10.1.1.70, class: telemetry, start: 5, size:  200, type: OnOff, protocol: udp, rate: 10kbps}

  - csvfile: case1/sat_53_d.csv
    address: 10.1.1.70
    antenna: {type: bicosine-backlobe, beamwidth-az: 30, beamwidth-el: 30, offset-az: 180, offset-el: 0}
    applications:
      - {destAddress: 10.1.1.60, class: telemetry, start: 150, size:  200, type: OnOff, protocol: udp, rate: 10kbps}
      - {destAddress: 10.1.1.50, class: telemetry, start: 150, size:  200, type: OnOff, protocol: udp, rate: 10kbps}
      - {destAddress: 10.1.1.40, class: telemetry, start: 150, size:  200, type: OnOff, protocol: udp, rate: 10kbps}
```


Figure A.2: **Network configuration file for Application 1 (Part 2 of 2)**.

Appendix B

Additional Information for Application 2

This section shows the full network config file used in Application 2.

```yaml
# Application 2: Cross-Plane TCP and UDP

VERSION: 0.1

config:
  startTime:          2673
  runTime:            480
  seed:               1     # random seed for simulation; cannot be zero

# configuration parameters for output tracing
tracing:
  prefix:             ./
  animFileName:       sns3-sim.xml
  traceFileName:      sns3-sim.tr
  routeFileName:      sns3-sim.routes
  neighFileName:      sns3-sim.neighbors
  flowFileBase:       flow_stats
  linkFileBase:       link_stats
  logInterval:        1.0
  verboseNet:         false
  verbose:            false
  specAnalyzerNode:   null

network:
  addressbase:        10.1.1.0
  bcastaddress:       10.1.1.255
  UDPPort:            7100
  TCPPort:            7200
  routing:            olsr
  routingAttributes:
    HelloInterval:    0.1

radio:
  standard:           802.11a
  staManager:         ns3::MinstrelWifiManager
  staManagerAttributes: null
  mac:
    Slot:                          36us   # 18us    # slot time
    Sifs:                          30us   # 20us    # SIFS duration
    AckTimeout:                    264us  # 176us   # ACK timeout duration
    CtsTimeout:                    264us  # 176us   # CTS timeout duration
    Rifs:                          6us    # 4us     # RIFS duration
    BasicBlockAckTimeout:          6us    # 4us     # Basic Block ACK timeout duration
    CompressedBlockAckTimeout:     336us  # 224us # Compressed Block ACK timeout duration
  dataMode:           OfdmRate6Mbps
  txPwrDbm:           20
  txPwrDbmMax:        30
  txPwrDbmMin:        0
  txPwrLevels:        31
  frequencyMHz:       2412
# ccaThreshold:       -50
```

Figure B.1: **Network configuration file for Application 2 (Part 1 of 2)**.

```yaml
events:
  - {type: wifiRadioOn, address: 10.1.1.20, start: 0}
  - {type: wifiRadioOn, address: 10.1.1.10, start: 0}
  - {type: wifiRadioOn, address: 10.1.1.30, start: 0}
  - {type: wifiRadioOn, address: 10.1.1.70, start: 120}
  - {type: wifiRadioOn, address: 10.1.1.60, start: 240}
  - {type: wifiRadioOn, address: 10.1.1.50, start: 240}
  - {type: wifiRadioOn, address: 10.1.1.40, start: 240}

nodes:

  - csvfile: case1/sat_48_b.csv
    address: 10.1.1.10
    antenna: {type: bicosine-backlobe, beamwidth-az: 30, beamwidth-el: 30, offset-az: 0, offset-el: 0}
    applications:
      - {destAddress: 10.1.1.30, class: download,  start: 0, size:  400, type: BulkSend, bytes: 1e9}
      - {destAddress: 10.1.1.30, class: telemetry, start: 0, size:  200, type: OnOff, protocol: udp, rate: 10kbps}

  - csvfile: case1/sat_48_c.csv
    address: 10.1.1.20
    antenna: {type: bicosine-backlobe, beamwidth-az: 30, beamwidth-el: 30, offset-az: 0, offset-el: 0}
    applications:
      - {destAddress: 10.1.1.30, class: download,  start: 0, size:  400, type: BulkSend, bytes: 1e9}
      - {destAddress: 10.1.1.30, class: telemetry, start: 0, size:  200, type: OnOff, protocol: udp, rate: 10kbps}

  - csvfile: case1/sat_48_a.csv
    address: 10.1.1.30
    antenna:
      - {type: bicosine-backlobe, beamwidth-az: 30, beamwidth-el: 30, offset-az: 0, offset-el: 0}
      - {type: bicosine-backlobe, beamwidth-az: 30, beamwidth-el: 30, offset-az: 90, offset-el: 0}
    applications:
      - {destAddress: 10.1.1.70, class: telemetry, start: 120, size:  200, type: OnOff, protocol: udp, rate: 10kbps}
      - {destAddress: 10.1.1.70, class: download,  start: 120, size:  400, type: BulkSend, bytes: 2e9}

  - csvfile: case1/sat_53_d.csv
    address: 10.1.1.70
    antenna: {type: bicosine-backlobe, beamwidth-az: 30, beamwidth-el: 30, offset-az: 180, offset-el: 0}
    applications:
      - {destAddress: 10.1.1.40, class: telemetry, start: 240, size:  200, type: OnOff, protocol: udp, rate: 10kbps}
      - {destAddress: 10.1.1.40, class: download,  start: 240, size:  400, type: BulkSend, bytes: 1e9}

  - csvfile: case1/sat_53_c.csv
    address: 10.1.1.60
    antenna: {type: bicosine-backlobe, beamwidth-az: 30, beamwidth-el: 30, offset-az: 180, offset-el: 0}

  - csvfile: case1/sat_53_b.csv
    address: 10.1.1.50
    antenna: {type: bicosine-backlobe, beamwidth-az: 30, beamwidth-el: 30, offset-az: 180, offset-el: 0}

  - csvfile: case1/sat_53_a.csv
    address: 10.1.1.40
    antenna: {type: bicosine-backlobe, beamwidth-az: 30, beamwidth-el: 30, offset-az: 180, offset-el: 0}
    applications:
      - {destAddress: 10.1.1.20, class: download,  start: 360, size:  400, type: BulkSend, bytes: 1e9}
```

Figure B.2: **Network configuration file for Application 2 (Part 2 of 2).**