RELIABILITY ANALYSIS OF LOW EARTH ORBIT BROADBAND SATELLITE COMMUNICATION CONSTELLATIONS

by

Islam Aly Sadek Nazmy

A Thesis

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Master of Science

School of Aeronautics and Astronautics

West Lafayette, Indiana

August 2020

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Karen Marais, Chair

School of Aeronautics and Astronautics

Dr. Daniel DeLaurentis

School of Aeronautics and Astronautics

Dr. Ali Raz

School of Aeronautics and Astronautics

Approved by:

Dr. Gregory Blaisdell

ACKNOWLEDGMENTS

Thank you to all the members of my committee for providing me with insightful advice throughout my master's thesis. Thank you to my advisor, Professor Marais, for her continued support and guidance during my master's degree.

I also want to thank my friends and family for supporting me and motivating me through to the end. Thank you to my mom and dad, for always standing behind me through everything.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Large space-based communication networks have been growing in numbers of satellites, with plans to launch more than 10,000 satellites into Low Earth Orbit (LEO). While these constellations offer many advantages over ground-based communication systems, they pose a significant threat when they fail and generate space debris. Given the reliability of current satellites, engineers can use failure modeling to design satellite constellations that are more resilient to satellite failures. Several authors have analyzed the reliability of geostationary satellites, but few have expanded the work to multiple-satellite systems.

To address this gap, we constructed a simulation model to show the performance of satellite constellations with different satellite reliability functions over time. The simulation model is broken down into four key parts: a satellite constellation model, a network model, a failure model, and a performance metric. We use a Walker star constellation, which is the most common constellation for LEO broadband satellite constellations. The network consists of satellite-to-satellite connections and satellite-to-groundstation connections, which routes data using a shortest-path algorithm. The failure model views satellites as either operational or failed (no partial failures) and considers the groundstation operator's knowledge or lack thereof of the satellites' operational status and uses satellite reliability to estimate the expected data throughput of the system. We also created a performance metric that measures how well the entire network is operating and helps us compare candidate constellations.

We used the model to estimate performance for a range of satellite reliabilities, and for groundstations with different numbers of communication dishes (effectively, satellite-ground links). Satellite reliability is a significant contributing factor to the long-term constellation performance. Using the reliability of small-LEO satellites, we found that a constellation of 1,200 small-LEO satellites completely fails after less than 30 days, given that we do not consider partial failures. Satellite constellations with higher satellite reliability, such as large geostationary satellites, last less than 50 days. We expect the constellations in our model to perform worse than real satellite systems, since we are only modeling complete failures, however these findings provide a useful worst-case baseline for designing sustainable satellite constellations. We also found that the number of groundstation-to-satellite communication links at each groundstation is not a significant factor for more than five communication links, meaning that adding more

communication antennas to existing satellite groundstations would not improve constellation performance significantly.

1. INTRODUCTION

1.1 Motivation

Communication satellites are an essential part of telecommunication networks for military, commercial, and scientific applications. Space-based broadband communication networks offer several advantages over traditional fiber-optic cable networks including wider coverage area, faster communication links, and cheaper cost of implementation. The demand for broadband satellite communication has driven companies to design large space-based communication constellations, termed mega-constellations. These mega-constellations are planned to have thousands of satellites in Low Earth Orbit (LEO), however most are not fully implemented yet. LEO is becoming more crowded with orbital debris, which raises the concern of debris from mega-constellations. The more satellites we launch into orbit, the more likely collisions are, generating yet more debris. Mega-constellations in LEO offer many advantages, however engineers should be cautions when designing such large constellations.

We need only to look at the failed constellations of the last few decades to see that poorly designed mega-constellations can have long-lasting repercussions. Iridium, Globalstar, and several other constellations were designed to meet the rising demand for space-based communication networks, yet many filed for bankruptcy. Iridium Communications found that almost 30 percent of their satellite fleet failed in orbit, despite the fact that they used “highly reliable components” (Foust, 2019), and filed for bankruptcy in 1999. Uncontrolled satellites in populous orbit regimes can collide with operational satellites or other space debris, exacerbating the threat of debris collision. Amazon reported to the U.S. Federal Communications Commission (FCC) that if 15 percent of their 3,236-fleet Kuiper System broadband internet satellites fail, there is a 12 percent chance that one of those failed satellites would collide with a piece of space debris (Harris, 2019). The message is clear: ignoring the effects of satellite failures when designing mega-constellations can lead to detrimental consequences.

Rather than using thousands of satellites for resilient broadband networks, we are interested in investigating the effect that satellite reliability has on the long-term performance of broadband satellite constellations. It is counter-intuitive to address the risk of satellite failures by proposing larger constellations, as this will increase the total impact of satellite failures. By modeling a

constellation design with satellite failures, engineers can make more informed decisions on how to design resilient constellations without using an excessive number of satellites.

1.2 Approach

We can think of a satellite constellation as a system, where multiple components interact with each other to produce a desired output. The system's performance is evaluated based on the desired system output. In an ideal system, where components do not fail, the system's performance remains constant over time. The sample system schematic in Figure 1.1 illustrates an ideal system's performance with no failures.

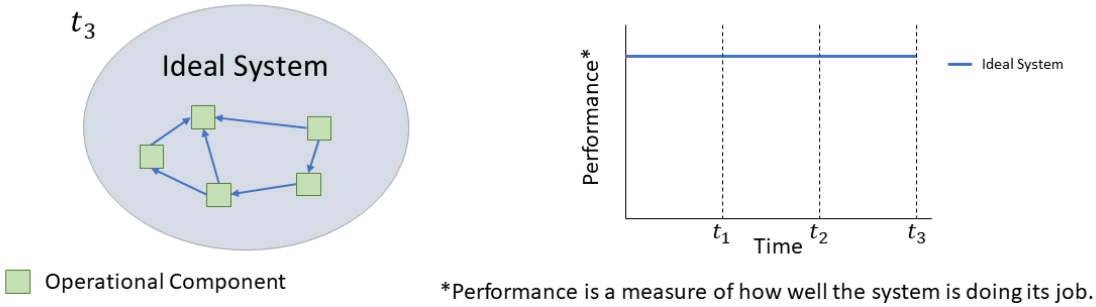

Figure 1.1: Sample system diagram with operational, interacting components (left) and the measured system performance metric of the ideal system (right)

In a real system however, individual components fail. These failures occur from improper use, component deterioration, or external forces. As components fail fewer interactions between system components produces a lower output. A sample system in Figure 1.2 shows a real system, with failed components, and the measured performance over time.

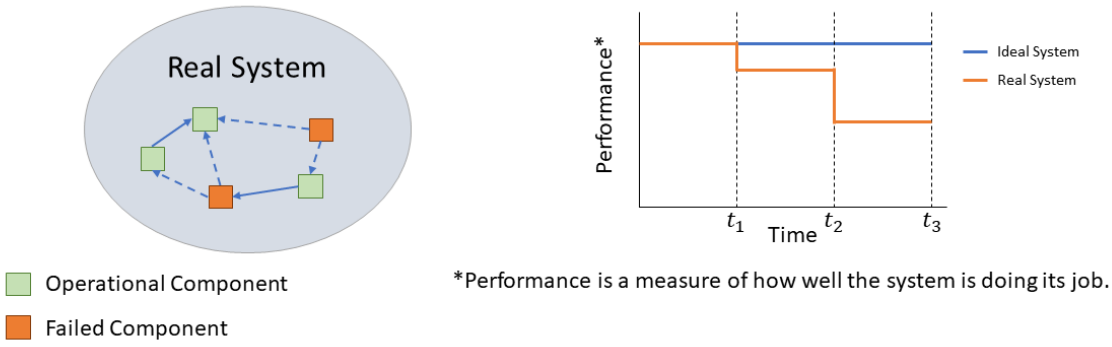

Figure 1.2: Sample non-ideal system with failed components where dashed arrows indicate a lost interaction due to component failures (left) and the system's performance metric over time (right)

In this thesis, we create a model of satellite constellations that considers the impact on overall performance with satellite failures. Engineers can use this model to help design constellations that are resilient to inevitable satellite failures with respect to the communication network's performance.

Our approach to design a useful model for broadband communication networks is comprised of four key elements: (1) the satellite constellation model; (2) the communication network model; (3) the satellite failure model; and (4) a definition of the network performance metric.

The satellite constellation model consists of the definition of the satellites' orbits, the propagation method for the satellites, and the groundstation positions for the constellation. Using a generic definition of satellite constellations' orbits, we can propagate each of the system's components to their new positions. Unlike geostationary orbits, which maintain a satellite at one position relative to the Earth's rotating frame, nearly all satellite constellations use non-geostationary orbits where the satellites are not fixed in the Earth rotating frame. This means that the satellites and groundstations are not fixed relative to each other. Due to both the satellite motion and failing satellites, the output performance of a satellite communication network cannot be reduced to a closed-form analytical solution. We thus need to propagate the satellites and compute the network at each timestep.

The dynamic system means that satellites are coming in and out of the field of view of groundstations, changing the overall network topology. We define a communication network model which outlines the connectivity rules—which system nodes connect to other system nodes

in certain conditions. By designing a set of communication link rules, we recompute the network topology at each timestep. The communication network model also defines the volume of data sent between groundstations. Without any fully constructed broadband communication networks, we rely on predictions of future groundstation positions from the literature.

The critical part of the satellite broadband communication network model is the satellite failure model. The satellite failure model is what brings realism to the ideal model, and thus greater foresight for constellation designers. We assume that satellite failures are non-recoverable, which is common for the relatively low-cost LEO satellites used in mega-constellations. Using a statistical model of satellite reliability, we simulate satellite failures at each simulation timestep. This allows us to model when which satellites fail, and how the overall network performance behaves over time.

Lastly, we define a performance metric which measures how well the communication network is functioning as satellites fail over time. The performance metric is based on how much data the network is delivering to groundstations, with component failures. We compare the performance of a network with satellite failures to a perfect network which connects all groundstations with no failures. This measure allows us to compare different constellation designs and different satellite reliabilities.

1.3 Thesis Outline

This thesis explains the background and method for creating a broadband satellite constellation model with satellite failures. In Chapter 2 we discuss the advantages and disadvantages of different orbit regimes and select a common constellation architecture for our model. In Chapter 3 we explain how the individual network components are connected to route data between groundstations. In Chapter 4 we examine how previous work has approached the problem of modeling satellite failures in constellation networks, and we outline a method to model satellite failures in our model. In Chapter 4 we also present a performance metric that quantifies the total network data throughput with failures, compared to the ideal network throughput without any failures. In Chapter 5 we lay out a baseline simulation case and address some of the limitations of this simulation method in practice, and also conduct a sensitivity analysis of the model, which highlights some of the important variables to consider when designing a satellite constellation. Chapter 6 concludes the thesis and identifies areas of future work.

2. SATELLITE CONSTELLATION MODEL

2.1 Satellite Constellation Background

A satellite constellation is defined as a number of similar satellites, of similar type and function, designed to be in similar, complementary orbits for a shared purpose, under shared control (Wood, 2001). There are many different types of communication constellation satellite architectures, each applicable to different types of satellite communication networks. Satellite constellation networks began by using a bent-pipe architecture with geostationary satellites for TV broadcasting, where a single satellite relays data from one groundstation to another groundstation. Satellite constellations for broadband data require much larger fleets of satellites, for which the traditional bent-pipe architecture is not economically viable (Farserotu & Prasad, 2000).

The altitude of satellites in a constellation is a significant factor in determining the number of satellites to use in a constellation (Wood, 2001). Higher altitude orbits can provide a greater coverage area, thus require less satellites overall. At an altitude of 35,800 km above the equator, geostationary satellite constellations can cover the entire Earth's surface with a minimum of three geostationary satellites which has made them popular for servicing large coverage areas (Wood, 2001). Although geostationary satellites can deliver coverage to vast regions of the Earth, they are expensive to launch and have significant latency of ~120 ms. Additionally, the geostationary ring is crowded with functional satellites and space debris, making them undesirable orbits for large broadband satellite constellations (Jehn et al., 2005).

Medium Earth Orbits (MEO), at an altitude between 9,000 km and 11,000 km, are cheaper to reach and still cover a relatively large footprint. Propagation delay for the uplink and downlink between groundstations and the satellites is approximately 30–37 ms, which is significantly less than the latency for geostationary orbits. The Hughes Spaceway non-geostationary (NGSO) proposal takes advantage of the smaller required satellite fleet in MEO, however no operational broadband constellations have opted to use MEO regimes due to the higher latency compared to Low Earth Orbits (Wood et al., 2001).

Low Earth Orbits (LEO) are the popular choice of orbits for broadband satellite constellations. SpaceX, Telesat, Kepler Communications, and at least eight other companies have begun construction, or filed for permits to the FCC, for satellite constellations that use

predominantly LEO orbits (Schneiderman, 2019). LEO is defined as an orbit with an altitude of 160 km to 2,000 km. At these altitudes, the one-way communication latency is 0.5 ms to 7 ms. Although the footprint of these orbits is extremely small, companies have opted to use more satellites to take advantage of the lower communication latency and cheaper cost to launch. SpaceX has been approved by the FCC to launch 7,518 satellites into a Very Low Earth Orbit (VLEO), operating below 350 km altitude. Satellites in the VLEO regime face the additional obstacle of combating atmospheric drag, which is a significant factor for satellites below 500 km altitude.

Within the LEO regime, there are several different constellation topologies that are appropriate for broadband satellite constellations. There are many satellite constellations that use multiple orbit altitudes, but most constellation network designs use regular constellations. Regular constellations are constellations composed of orbital planes with the same altitude and orbital inclination, which simplifies satellite tracking for groundstations (Wood, 2003). The two major LEO regular constellation types used in practice are: (1) Walker star constellations; and (2) polar constellations. Both Walker star constellations and polar constellations have circular orbital planes of the same altitude with equally spaced satellites within each plane by their mean anomaly, and equally spaced orbital planes within the plane of reference by their right ascension.

Walker star constellations are a subset of the larger family of Flower Constellations. Walker star constellations are distinct from polar constellations in that their orbital inclination is greater than 85° (Wang et al., 2006). These constellations get their name from the star shape they make when viewed from polar regions, as shown in Figure 2.1 (Mortari et al., 2004).

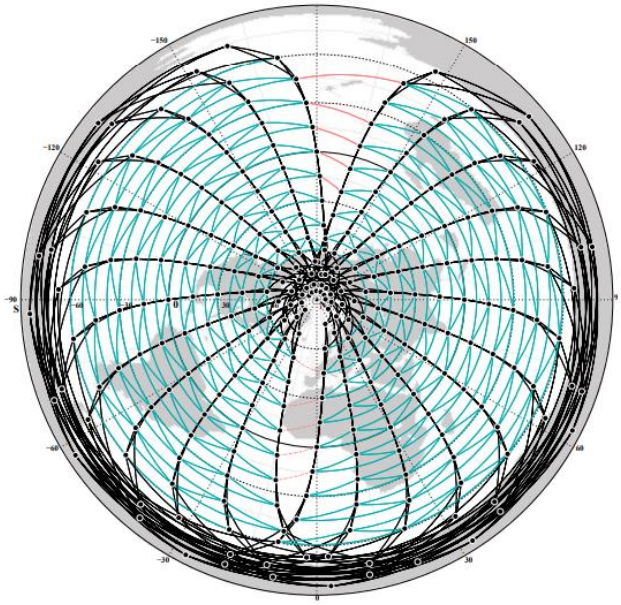

Figure 2.1: Teledesic's (Boeing design) Walker star constellation using an 82.3° inclination, viewed from the North Pole (Mortari et al., 2004).

Polar constellations are reserved for regular constellations which have an orbital inclination greater than 85°85°. These constellations provide coverage at higher latitudes, however launching satellites to polar orbits is significantly more expensive due to the large amount of fuel required to change inclination. OneWeb uses a polar constellation configuration, as shown in Figure 2.2.

Figure 2.2: One Web's polar constellation design using an 87.9° inclination, viewed from the equatorial plane (del Portillo et al., 2019)

2.2 Satellite Constellation Parameterization

For our broadband constellation model, we consider Walker star constellations due to their prevalence in industry (del Portillo et al., 2019). The total number of satellites in the Walker star constellation network is $M_L \times N_L$, where M_L is the number of identical orbital planes with the same inclination, and N_L is the number of satellites in each plane. The first orbital plane in the constellation is defined from an initial right ascension of Ω_0 from Vernal Equinox, and the first satellite within each orbital plane is defined from an initial mean anomaly M_0 from the orbital plane's ascending node. We denote satellites in different orbital planes with the same mean anomaly as *similar* satellites.

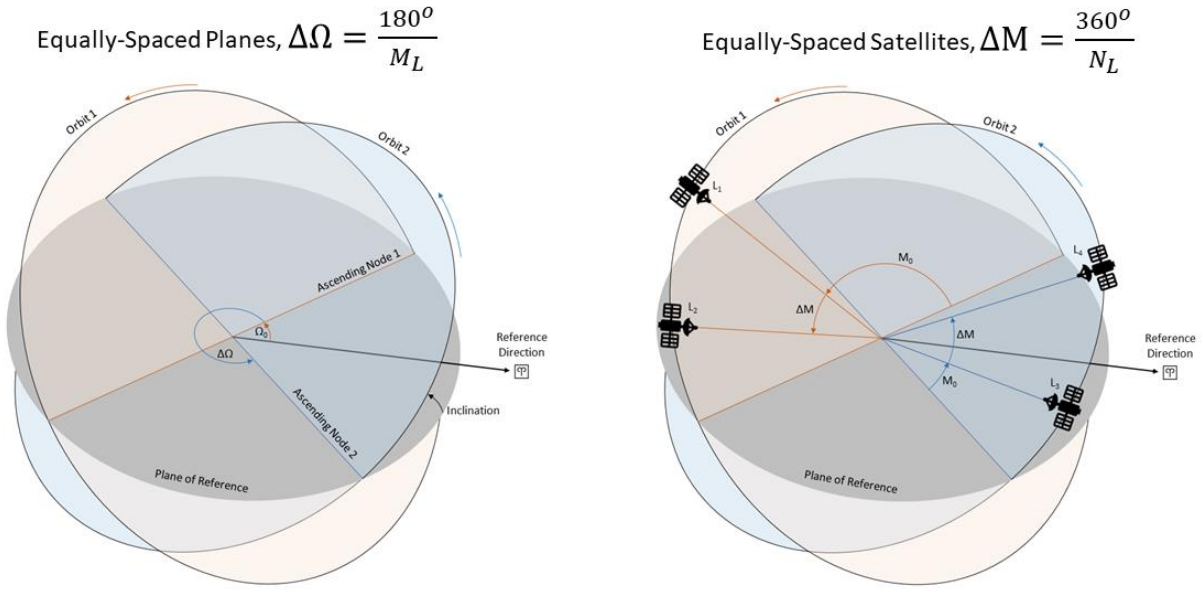

Figure 2.3: Schematic of regular constellations with equally spaced planes by their right ascension in the (left), and equally spaced satellites within the same plane (right)

To fully define the Walker star constellation, we also define the semimajor axis of each orbit a, the orbit inclination i, and the starting epoch t_0. A Walker star constellation can thus be generically described as $W \in \{M_L \times N_L : a, e, i, \Omega_0, M_0, t_0\}$.

Table 2.1: Summary of satellite constellation parameterization and parameter domains

Term	Definition
M_L	Number of orbital planes in the constellation
N_L	Number of satellites in each orbital plane
a	Semimajor axis of the orbits [180 km, 2000 km]
i	Inclination angle of the orbits (0°, 85°)
Ω_0	Right ascension of the ascending node of the first plane at epoch [0°, 360°)
M_0	Mean anomaly of the first satellite in the orbital planes at epoch [0°, 360°)
t_0	Starting epoch in Julian Day (JD)

2.3 Dynamics Model

From the initial conditions at the starting epoch defined in Table 2.1, we choose to propagate the satellites using a two-body model and not considering attitude dynamics. There are significant forces in the LEO regime that perturb the two-body motion, most significantly are the $J_{2.0}$ spherical

harmonics and atmospheric drag, however we assume that the constellation satellites have station-keeping capabilities to counter these perturbing forces.

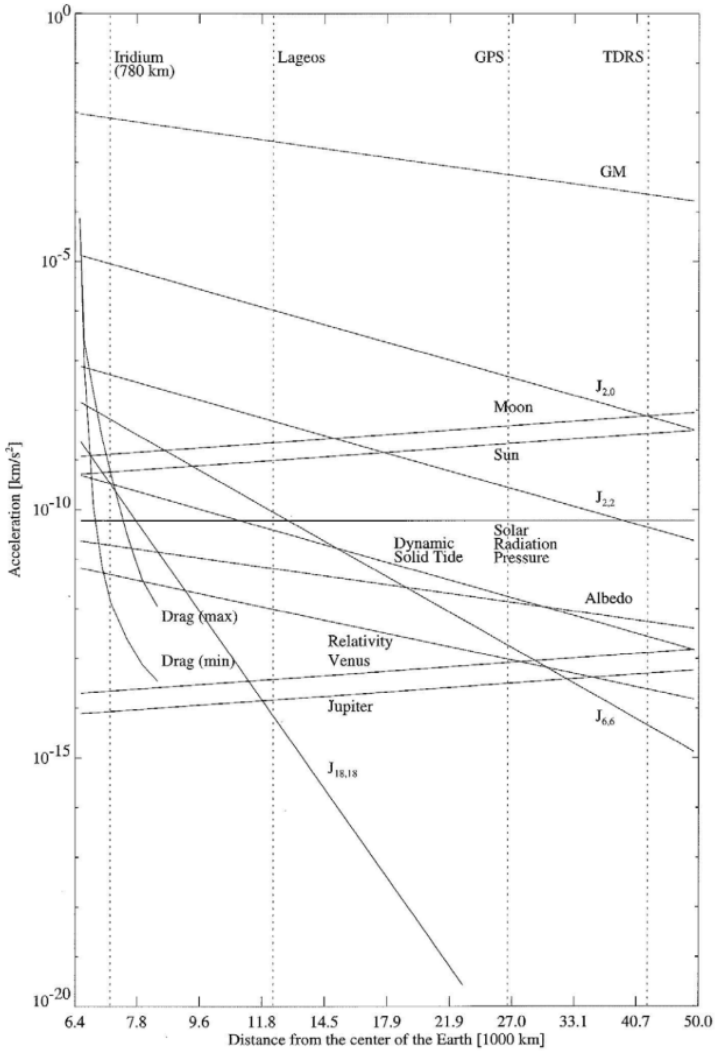

Figure 2.4: Acceleration forces acting on satellites in Earth orbit varying with orbital radius, where J_i indicates the i^{th}-degree spherical harmonics (Howell, 2018)

The equations of motion for the two-body problem can be expressed as three second-order differential equations, where (x, y, z) is the spacecraft's position in the inertial frame, $(\dot{x}, \dot{y}, \dot{z})$ is the spacecraft's velocity in the inertial frame, and μ is the Earth's gravitational constant.

$$\ddot{x} = -\frac{\mu}{\sqrt{x^2 + y^2 + z^2}}x \tag{2.1}$$

$$\ddot{y} = -\frac{\mu}{\sqrt{x^2 + y^2 + z^2}}y \tag{2.2}$$

$$\ddot{z} = -\frac{\mu}{\sqrt{x^2 + y^2 + z^2}}z \tag{2.3}$$

2.4 Summary

In this chapter we discussed the advantages of different orbital regimes for broadband data communication, and we pointed out two of the commonly used satellite constellation types for large LEO mega-constellations. We selected a Walker star constellation to create our simulation model and summarized the defining constellation parameters in Table 2.1. We decided to ignore orbital perturbations for this dynamics model since active satellites use station-keeping maneuvers to maintain two-body orbits.

3. NETWORK MODEL

3.1 Satellite Communication Background

Communication links in satellite constellations can be categorized as groundstation-to-satellite communication links, satellite-to-groundstation communication links, and satellite-to-satellite communication links. Each of these communication links operate with a specific radio frequency (RF) to overcome the obstacles of each data link environment. The most common RF band for communication satellite constellations is the K-band, which is further categorized into K_u and K_a bands.

Table 3.1: Radio frequency bands with corresponding wavelength and frequency range (Spencer, 2019; del Portillo et al., 2019; OneWeb, 2013)

Band	Wavelength Range (cm)	Frequency Range (GHz)
Ultra-high frequency (UHF)	100–10	0.3–3
L	30–15	1–2
S	15–7.5	2–4
C	7.5–3.75	4–8
X	3.75–2.4	8–12
K	2.4–0.75	12–40
K_u	2.5–1.67	12–18
K_a	1.11–0.75	26.5–40
Q	0.75–0.6	40–50
V	0.6–0.4	50–80
W	0.4–0.3	80–90

Groundstation-to-satellite communication links are often the most robust data links since transmitter power from the groundstation can be much higher than that from a satellite. The groundstation-to-satellite link uses a higher radio frequency, which improves the signal-to-noise ratio (Spencer, 2019).

Electromagnetic radiation is absorbed by several gases in the Earth's atmosphere, most significantly due to water vapor (H_2O), carbon dioxide (CO_2), and ozone (O_3), which limit the RF bands for groundstation-to-satellite and satellite-to-groundstation communication. Signals between satellites and ground segment nodes use radio waves, which are not absorbed by atmospheric gasses (Humboldt State University, 2014).

Nearly all broadband communication satellites use K-band communication links for groundstation-to-satellite and satellite-to-groundstation communication (del Portillo et al., 2019). The K-band wavelength is long enough to pass through the Earth's atmosphere while still offering a high frequency for a high signal-to-noise ratio.

The minimum elevation angle also influences the atmospheric interference in groundstation-to-satellite and satellite-to-groundstation communication links. The elevation angle, θ_h, is the satellite's angle above the local horizon from a groundstation. The minimum elevation angle for a satellite to communicate with a groundstation (and vice versa) limits the slant range between the satellite and the groundstation, and thus the interference due to the atmosphere (Li and Liu, 2002).

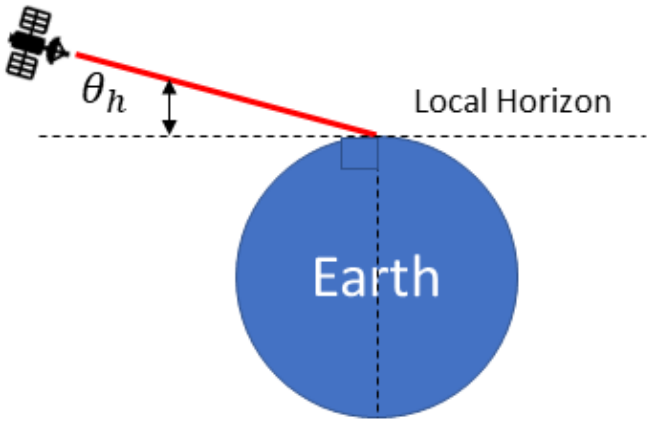

Figure 3.1: 2-D geometry of the satellite elevation angle θ_h from a groundstation

Satellite-to-satellite communication link bands connect two satellites within an unobstructed line of sight. Although these communication links are not restricted by the atmospheric absorption spectrum, they are restricted by the satellite antenna's pointing error. Both the sending and receiving satellites must point their steerable transmitter and receiving antennas, respectively, at each other for successful satellite-to-satellite communications. Satellites within the same circular orbital plane are fixed relative to each other, however satellites in different planes move relative to each other, which requires active tracking and gimballing of the antennas. Near the poles, the relative speed between satellites in different planes is so great that the satellite antennas often cannot track satellites in adjacent planes (del Portillo et al., 2019). One solution to this problem is using omnidirectional antennas, which do not require mechanical steering; however, they are expensive and therefore uncommon in LEO broadband satellite constellations. To avoid data

packet loss in cross-plane satellite-to-satellite communication links, constellation designers disconnect the satellite-to-satellite communication links when at least one satellite crosses a predetermined critical latitude near the poles (del Portillo et al., 2019; Li and Liu, 2002).

3.2 Ground Segment Topology Background

Satellite communication constellations are designed to connect two groundstations through a data link that passes through one or more satellites; the capabilities of the groundstations are a major factor in determining the type of network topology to be used. Satellite communication networks can be categorized into two basic network topologies: (1) hub/remote networks; and (2) point-to-point networks (Ayers, 2012).

In a hub/remote network topology, a single hub groundstation serves as the central groundstation which transmits data to smaller remote groundstations. The hub/remote network topology is typically used for applications requiring asymmetric data exchange, such as remote internet access (Ayers, 2012). Figure 3.2 shows the hub/remote network, where the majority of data is sent from the hub groundstation to the remote groundstations. The hub station uses a large-aperture antenna to transmit data to the satellite constellation, while remote stations use very small aperture terminals (VSATs) for two-way communication with satellites. In a hub/remote network topology, the hub groundstation is the most important node of the network (Ayers, 2012). Unlike the remote groundstations, a failure of the hub groundstation would cause the entire network to fail.

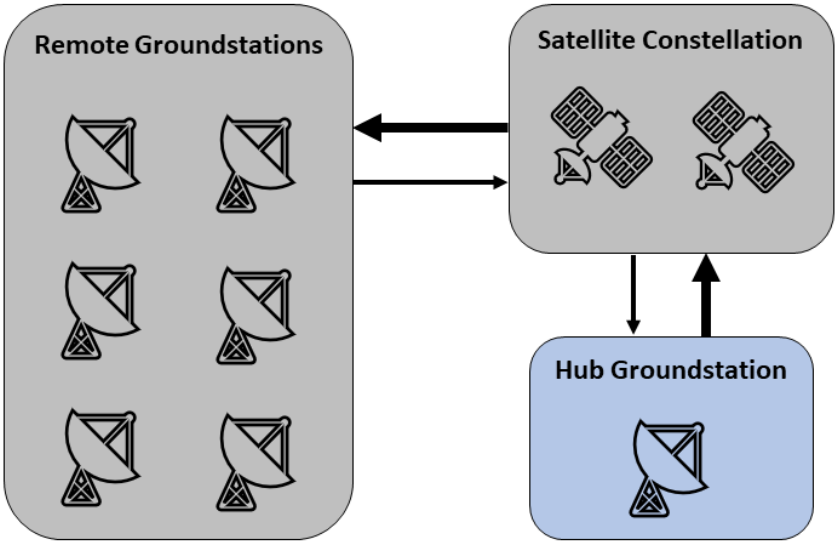

Figure 3.2: Hub/remote communication constellation topology, where the thickness of the black arrows is proportional to the volume of flow of data

Most communication satellite constellations use a point-to-point network topology, where communication exists between any two groundstations (Ayers, 2012). No single groundstation failure would cause the entire network to fail. We illustrate this network topology in Figure 3.3.

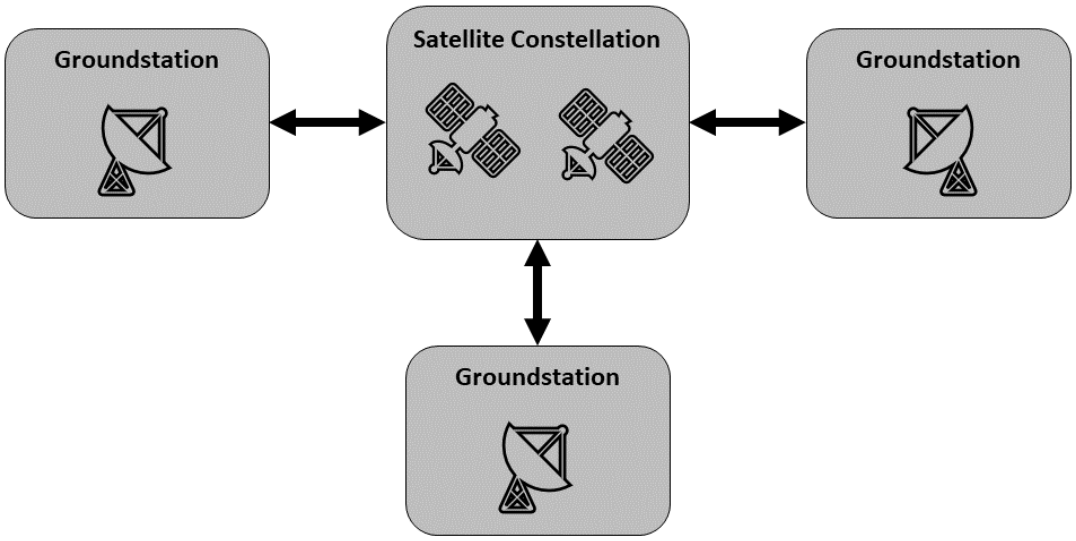

Figure 3.3: Point-to-point communication constellation topology, where the groundstations equally transmit to and receive data from other groundstations through the satellite constellation

3.3 Network Space Segment Model

We designed the network model from the Walker star constellation which is used in broadband satellite constellations such as Starlink, LeoSat, and Teledesic, and several others (del Portillo et al., 2019). We make the following assumptions about the constellation:

1. Satellites can sense their own locations (Latitude, Longitude). Using the groundstation signals and an atomic clock, satellites are often able to pinpoint their position above the Earth with high accuracy. We make this assumption so that satellites can automatically point their antennas and establish contact with groundstations at the appropriate times.
2. All communication links are bi-directional. We make this assumption to model a point-to-point communication network (Figure 3.5), while simplifying the communication links for our model.
3. There are no errors in communication links. We will only consider satellite failures for this reliability analysis to simplify the model.

We followed the mesh grid structure of most Walker star constellations, where each satellite is connected to its neighboring satellites to create a mesh grid network of two-way communication links (del Portillo et al., 2019). At each timestep, we computed the network adjacency matrix using a rule-based algorithm. There are two types of satellite-to-satellite communication links: in-plane satellite links, which connect a satellite to the two nearest satellites in the same orbital plane; and cross-plane satellite links, which connect a satellite to the two nearest similar satellites in neighboring orbital planes. In-plane satellite communication links can be identified as the *fore* and *aft* links from a satellite. The fore link is defined as the in-plane satellite link in the direction of the satellite's velocity, and the aft link is defined as in the direction opposite to the satellite's velocity. The relative position between satellites in the same orbital plane does not change significantly for near-circular orbits, so these satellite links are easy to maintain with the pointing antennas onboard.

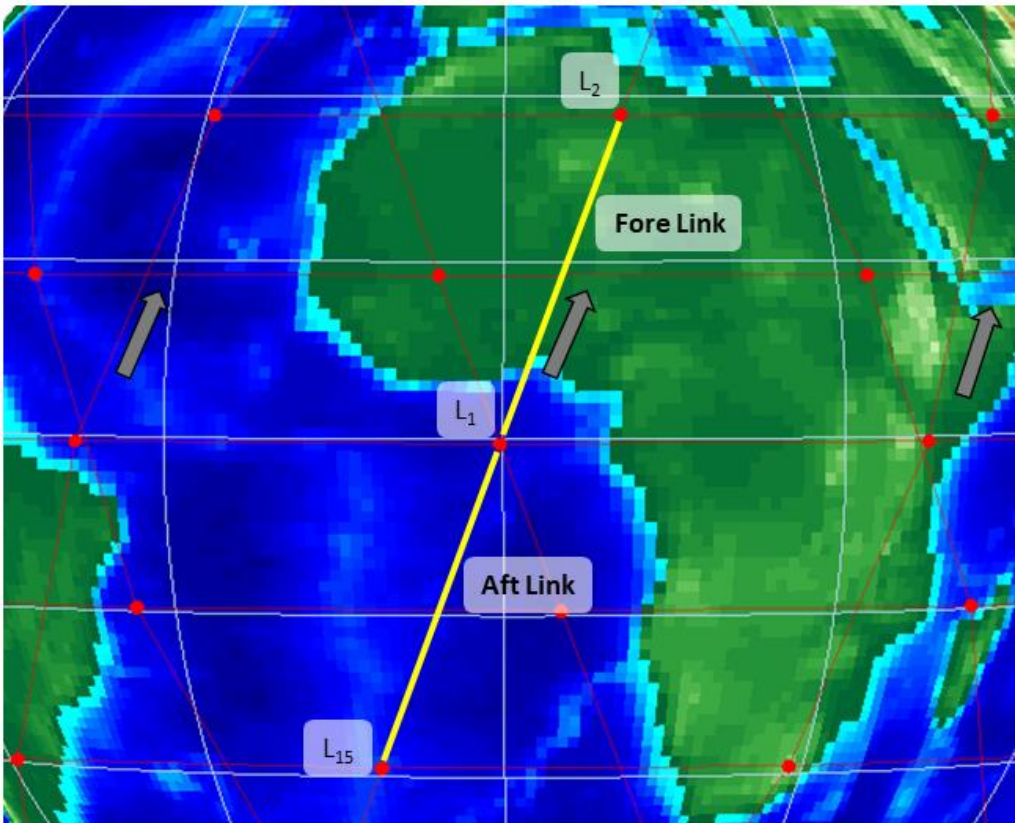

Figure 3.4: Schematic of in-plane satellite links for satellite L1 in a Walker Constellation $W \in \{15 \times 12: 7878\ \text{km}, 0°, 70°, 0°, 0°, JD\ 2458964\}$. The direction of each orbital plane velocity is indicated with grey arrows. Figure generated using NASA's General Mission Analysis Tool (GMAT).

The cross-plane links between a satellite and similar satellites in adjacent orbital planes are called the left and right links, defined as the port and starboard directions to the satellite, respectively. These satellite links are much harder to maintain, as the relative position between similar satellites changes significantly near the poles. We include cross-plane links in the overall network adjacency matrix when both satellites are below an absolute critical latitude β, or above the critical latitude -β. Constellations such as Iridium, Globalstar, and Teledesic have used critical latitudes from β=±40° to β=±75° (Wang et al, 2006). The yellow lines in Figure 3.7 highlight the cross-plane satellite links from satellite L3, while the red lines show all other satellite-to-satellite links. Note that for satellites above the 60° latitude, satellites do not maintain their cross-plane satellite links due to the critical latitude rule our model enforces.

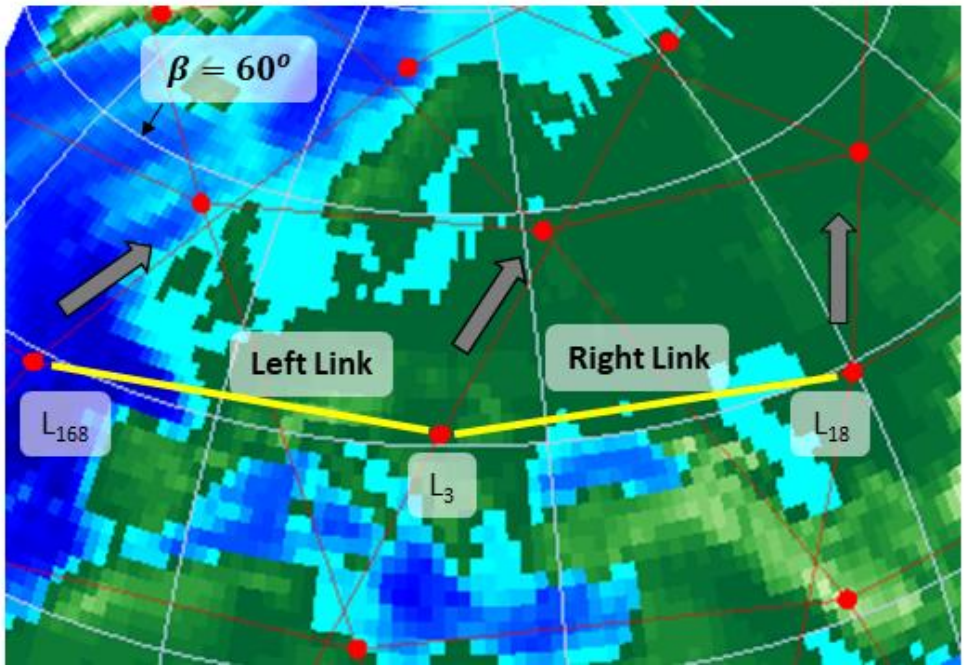

Figure 3.5: Schematic of cross-plane satellite links for satellite L3 in a Walker Constellation $W \in \{15 \times 12: 7878\ \text{km}, 0°, 70°, 0°, 0°, JD\ 2458964\}$, and a critical latitude $\beta = 60°$. The direction of each orbital plane velocity is indicated with grey arrows. Figure generated using GMAT.

3.4 Network Ground Segment Model

The last type of communication link is the ground station link: a two-way communication link between satellites and the ground station. While satellites can only maintain one groundstation link, ground stations can support more than four links simultaneously. We make the following assumptions for the groundstations:

1. Groundstations track satellites automatically. We do not explicitly model the satellite tracking algorithm or antenna pointing errors to simplify our model.
2. Groundstations connect to a maximum of 10 satellites nearest to the groundstation. This number was selected by reviewing the literature of ground segments, but it should be noted that there is great variation in the number of available antennas at groundstations in practice (del Portillo et al., 2018).
3. All groundstations are identical. We made this assumption to simulate the ground segment of a point-to-point network, where all groundstations uplink approximately the same amount of data that they downlink.

4. Satellites connect to a maximum of one groundstation at a time. Most LEO communication satellites only contact one groundstation at a time to simplify communication hardware.

We selected the locations of groundstations using a set of 77 candidate groundstation positions with plans to be constructed by various companies and agencies for broadband LEO constellations (del Portillo, 2018). There are currently no fully constructed ground segments for broadband constellations, however most constellations have plans to build 20–100 groundstations across the Earth (del Portillo et al., 2019). The selected groundstation locations are shown in Figure 3.6, and the list of groundstations can be found in Appendix A.

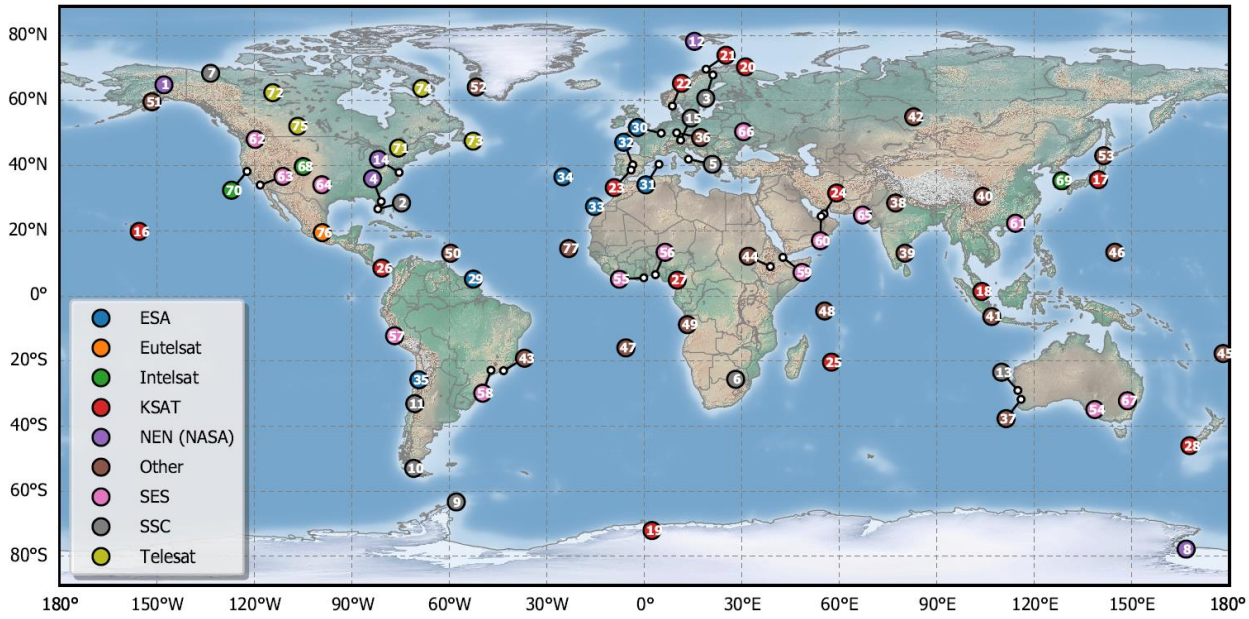

Figure 3.6: Candidate groundstation locations for our model (del Portillo et al., 2019)

3.5 Routing Algorithm

Given a satellite constellation network of fully connected nodes, a routing algorithm determines the path via which data will be routed to travel from one groundstation node to another. The transmitting groundstation assigns a routing path to each data packet, which cannot be changed once it leaves the groundstation (Ekici et al., 2001).

One of the most used routing algorithms for LEO constellations is the Datagram Routing Algorithm (DRA), which uses virtual nodes to route data through the network (Ekici et al., 2001). The DRA assumes that the entire Earth surface is covered by notional satellite locations evenly distributed across the Earth-fixed frame. The nearest satellite to each notional location assumes

the identity of that location and is used to route any data sent to that location. As satellites pass overhead, the satellite corresponding to a notional location changes. In this manner, the route of a path between groundstations does not change, however the satellites that are used to route the data can change. The routing algorithm rules is an iterative set of steps to ensure the fastest routing path between groundstations (Ekici et al., 2001). The algorithm operates on the principle that cross-plane satellite links are longer, and therefore slower, near the equator than they are near the poles.

The DRA is a multi-step process that requires approximating a data route using notional locations and refining the route according to the satellites' actual locations. This iterative process is useful in practice, but versions of the DRA algorithm vary greatly with the type of constellation and groundstation capabilities (Ekici et al., 2001; OneWeb, 2013). Because satellites in LEO travel so quickly, sometimes the nearest satellite to a notional location is only visible for a short period. Since groundstation communication dishes must gimbal to track satellites, it is sometimes more advantageous to assign satellites which are not nearest to the groundstations (Ekici et al., 2001), but will be in the field of view for longer. These constraints depend on satellite orbit altitudes and the limitations of groundstation communication dishes.

Implementing DRA is quite complex and specifically requires detailed orbital locations that are beyond the scope of this work. Therefore, to keep the focus on the main purpose of our work, we use a simpler shortest-path routing algorithm to calculate the data route between two groundstations. The shortest path routing algorithm uses the least number of intermediate satellite nodes in the network to route data between groundstations. According to Ekici et al. (2001), the DRA can be simplified to a shortest-path algorithm to reflect an approximate performance of the real routing algorithm (Ekici et al., 2001).

3.6 Summary

In this section, we explained the key methods for communication links in ground segments and space segments of constellation networks. We explained two common ground segment architectures and selected a point-to-point ground segment architecture for our simulation model. Satellites are connected to each other through a mesh-grid network of in-plane and cross-plane satellite communication links, which is the most common network topology of constellation space

segments. We also explained a significant data routing algorithm, the DRA, and how we simplified it to a shortest-path algorithm for our simulation model.

4. FAILURE MODEL

4.1 Satellite Failure Analysis Significant Works

Satellite failure modeling is a key area of research with a long history in geostationary satellite analysis. Since a geostationary satellite always remains at the same point in the sky, it can be thought of as a static system. Using the reliability function of a geostationary system, Saleh et al. (2005) created a failure model for estimating the net present value of satellites. Their net present value model of geostationary satellites only considers complete satellite failures, but it demonstrates a useful method which can be extended to other satellite systems. Saleh et al. derived the net present value of geostationary satellite systems with failure and compared this to the net present value assuming no failures.

Similarly, Ereau and Saleman (1996) designed a simulation for satellite constellations assuming complete, nonrecoverable failures based on Petri Nets. Their paper models the availability of satellite constellation networks with satellite failures based on the approach of the French Space Agency.

Saleh et al.'s, and Ereau and Saleman's, work assumes passive control, where the groundstations route data through the constellation without any prior knowledge of which satellites have failed. Ash and Newth (2007) tested different constellation network topologies against cascading failures, where the groundstation operators have some knowledge of which satellites have failed. Their work showed the impact that cascading failure has on satellite constellations, but more importantly it demonstrates the significance of groundstations' knowledge of satellite constellations on the overall performance of the network.

4.2 Failure Model Approach

The objective of this section is to outline our assumptions and key ideas to develop a failure model. Using the dynamical model of LEO satellites and the network model defining communication links between satellites and ground stations, we formulate a model of satellite failures with the following assumptions:

1. Satellite failures are fatal (not considering partial failures). This assumption simplifies the satellite failure model and is similar to other failure models in satellite reliability analysis (Saleh, 2005).
2. Satellite failures are independent. Most communication satellites are operationally independent, meaning that one satellite failure does not impact the reliability of other satellites. We do not consider satellite collisions with each other, or any other type of failure in which there is conditional dependence.
3. Satellite failures are identically distributed. We assume that all satellites are identical, and begin their operational lifetime at the same time, which simplifies our model.
4. Satellite operators can check if satellites are functional. We simulated the constellation where groundstation operators have some knowledge of the operational satellites in the constellation. If the operators know that a satellite in the constellation has failed, they will not attempt to route data through that failed satellite.
5. Groundstations do not fail. Groundstations can be maintained regularly, unlike satellites, which makes them significantly more reliable that satellites. We assume fully reliable groundstations to help simplify our model.
6. Communication links do not fail. We do not consider bit rate error or signal disruption due to noise in our simulation, to focus our study on the effects of satellite system failures on constellation performance.

In a simple satellite constellation where one groundstation is sending data to another groundstation, one or more satellite failures in a communication path will cause the entire path to fail. A failed path results in complete loss of the data being routed through the path. Once a data packet has been sent from a groundstation, the data packet will attempt to take the communication path assigned regardless of failed satellites. When a satellite along the path fails, the receiving groundstation does not receive the sent data, as shown in Figure 4.1.

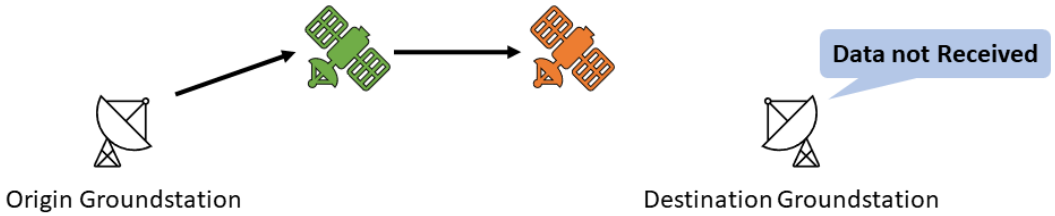

Figure 4.1: Schematic of a failed communication path between groundstations due to a failed satellite (red), showing the flow of data (black arrows)

The satellite constellation network can be abstractly thought of as a network made up of groundstation nodes and satellite nodes. In a simple scenario, two groundstation nodes can be connected through at least two possible data paths. Broadband constellations in practice are often large enough to offer multiple paths between groundstations; we want to capture this characteristic in our simulation. The schematic in Figure 4.2 shows two groundstations connected through two communication paths. Communication path *A* is the preferred path, while communication path *B* is an alternative path. Note that some groundstations may have more than two possible communication paths, but the same principle applies.

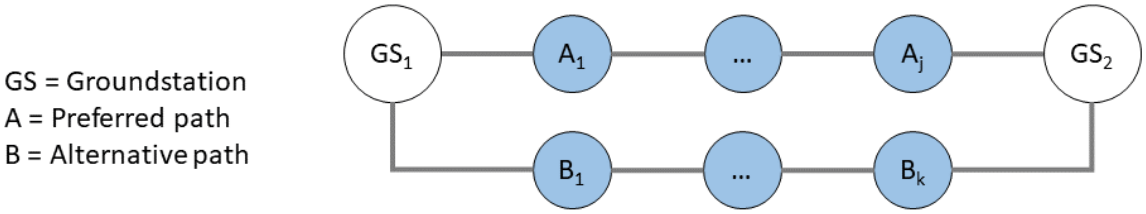

Figure 4.2: Nodal network of a generic satellite constellation, where circles are groundstations or satellites forming a communication path, and grey lines are active communication links

Following our assumption that groundstation operators have some knowledge about the functionality of satellites in the constellation, a satellite failure can result in one of two possible outcomes. The first possible outcome of a satellite failure is that a groundstation operator does not realize that a satellite along the primary path has failed and sends data along the primary path. We categorize this type of failure as an *unknown failure*, illustrated in Figure 4.3.

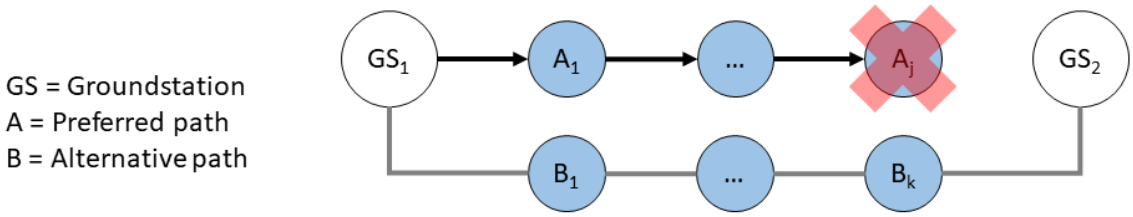

Figure 4.3: Nodal diagram of an unknown failure, where the groundstation operator does not reroute the data to the secondary path (B) when a satellite's primary path has failed (A)

The second possible outcome from a satellite failure is that the groundstation operator does realize that the preferred communication path has failed, and attempts to send the data along the alternative communication path. In this outcome, the data sent from the origin groundstation does reach the destination groundstation. We categorize this type of failure as a *known failure*, illustrated in Figure 4.4.

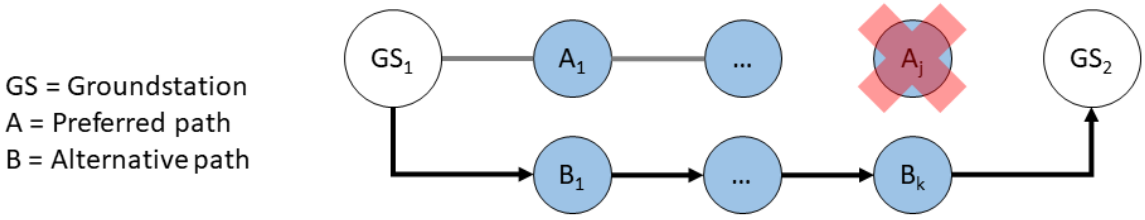

Figure 4.4: Nodal diagram of a known failure, where the groundstation operator actively reroutes the data from the primary communication path (A) to the secondary communication path (B)

A communication path between two groundstations must pass through at least one satellite. One or more satellites can fail along the path, with or without the groundstation operator's knowledge. We model unknown and known failures to create a failure model which simulates a broadband satellite network where the groundstation operators have some knowledge of satellite failures.

4.3 Unknown Failure Model Design

For the simple case of a communication path between two groundstations with one operational intermediate satellite, the unknown failure is the chance that the satellite will fail before

the next timestep. Assuming the intermediate satellite is operational at time t, we calculate the probability that the satellite will remain operational at the next time step, $t + \Delta t$. Assuming the satellite maintains communication with both groundstations at time $t + \Delta t$, the probability that the communication path will have failed at the time $t + \Delta t$ given that it was functional at t is expressed as:

$$Pr\big(t < T_f \leq t + \Delta t \,\big|\, T_f > t\big) \tag{4.1}$$

where T_f is the time to failure of the satellite. The probability that the satellite will fail in the next timestep, given that it had not failed at time *t* is:

$$Pr\big(t < T_f \leq t + \Delta t \,\big|\, T_f > t\big) = \frac{F(t + \Delta t) - F(t)}{R(t)} \tag{4.2}$$

Which can be simplified into a function of reliability using Equation 4.3. $R(t)$ is the reliability of the satellite at time t.

$$R(t) = 1 - F(t) \tag{4.3}$$

$$\begin{aligned} Pr\big(t < T_f \leq t + \Delta t \,\big|\, T_f > t\big) &= \frac{1 - R(t + \Delta t) - (1 - R(t))}{R(t)} \\ Pr\big(t < T_f \leq t + \Delta t \,\big|\, T_f > t\big) &= 1 - \frac{R(t + \Delta t)}{R(t)} \\ 1 - Pr\big(t < T_f \leq t + \Delta t \,\big|\, T_f > t\big) &= \frac{R(t + \Delta t)}{R(t)} \end{aligned} \tag{4.4}$$

where $1 - Pr\big(t < T_f \leq t + \Delta t \,\big|\, T_f > t\big)$ is the probability that the satellite has not failed at the time $t + \Delta t$ given that the satellite was operational at time t. This failure model can be illustrated using two-state failure model, as shown in Figure 4.5. A working satellite at time t can either remain functional at the next time step, or fail.

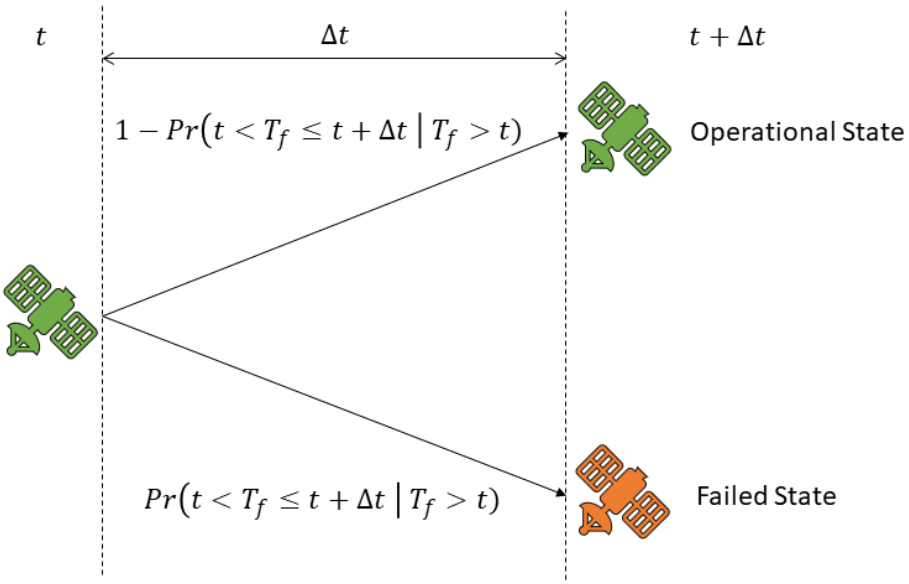

Figure 4.5: The possible states that an operational satellite at time t can transition to at time $t + \Delta t$, with the probability of transitioning to each state

Most communication paths use multiple satellites in series; we formulate a similar reliability of communication paths but with k identical intermediate satellites in the groundstation-to-groundstation communication path. The probability that the communication path will remain operational at time $t + \Delta t$ given that the communication path of k satellites is operational at time t can be expressed as a function of reliability, shown in Equation 4.5:

$$1 - Pr_k\left(t < T_f \leq t + \Delta t \mid T_f > t\right) = \left[\frac{R(t + \Delta t)}{R(t)}\right]^k \tag{4.5}$$

This analytical expression in Equation 4.5 describes the probability that a communication path will be operational at timestep $(t + \Delta t)$, given that it was operational at time t. While satellites in our model can either be functional or non-functional, represented by 1 and 0 respectively, the expected functionality is the average functionality of many systems with the same reliability function. This idea was used by Saleh et al. (2005) to estimate the net present value of a geostationary satellite system over time.

We implement the unknown failure model using the reliability of small LEO satellites (mass < 500 kg, apogee and perigee < 2000 km). Using fatal failure data, Saleh and Castet (2011) fitted a 2-Weibull mixture distribution to estimate the reliability of LEO satellites. The reliability curve

is only valid for 3.7 years, due to the limited empirical data available. The 2-Weibull mixture distribution can be expressed as:

$$R(t) = \alpha \exp\left[-\left(\frac{t}{\theta_1}\right)^{\beta_1}\right] + (1-\alpha)\exp\left[-\left(\frac{t}{\theta_2}\right)^{\beta_2}\right] \quad (4.6)$$

Table 4.1: The 2-Weibull mixture distribution parameters for reliability of small LEO satellites (Saleh and Castet, 2011)

Parameter	$\boldsymbol{\alpha}$	$\boldsymbol{\beta_1}$	$\boldsymbol{\beta_2}$	$\boldsymbol{\theta_1}$ **(years)**	$\boldsymbol{\theta_2}$ **(years)**
Value	0.9759	0.1822	12.4386	1167882337.03	3.45

The first Weibull shape parameter $\beta_1 < 1$ captures spacecraft infant mortality, while the second Weibull shape parameter $\beta_2 > 1$ captures the spacecraft wear-out failures.

Given k intermediate satellites in a groundstation-to-groundstation communication path, we can model the probability that a path will be operational at any time t using Equations 4.5 and 4.6. As the number of satellites in the communication path increases, the reliability of the path decreases. Plotting the reliability functions of paths with different numbers of intermediate satellites, k, Figure 4.6 shows that the reliability of paths with more intermediate satellites is lower than those with less intermediate satellites throughout the simulation period.

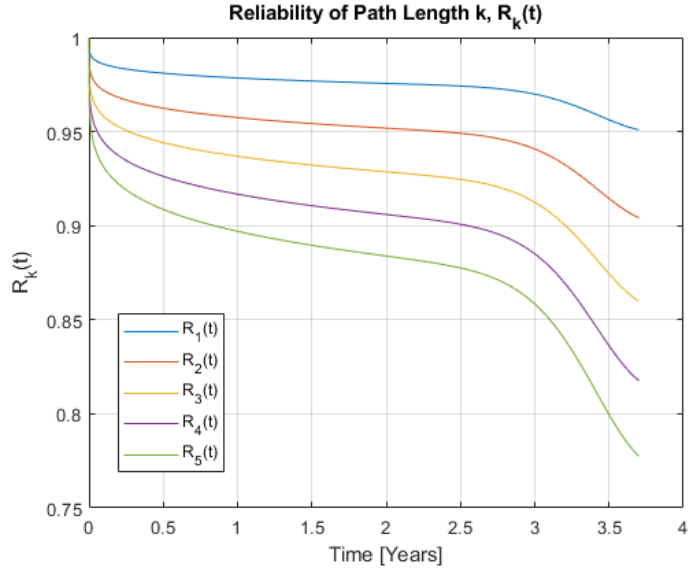

Figure 4.6: Reliability of a groundstation-to-groundstation communication path over time, $R_k(t)$, for k intermediate satellites

4.4 Known Failure Model

Until now, we have assumed that the satellite operators know the number of intermediate satellites in a path between ground stations. The number of intermediate satellites, k, determines the reliability of the communication path. In the unknown failure model, we assume that the path is operational at time t; however, if the operators know that a path is not operational at time t due to a failed satellite, they should select a different route to avoid the failed satellite. Rerouting the communication path can result in a different number of intermediate satellites k, and therefore a different path reliability.

While in the unknown failure model we could derive a closed-form expression to calculate the mean data throughput, the known failure model does not have an analytical solution. Consider a simple constellation with three satellites between two groundstations, as shown in Figure 4.7, where two groundstations are connected through either a single satellite, or a pair of satellites in series. If Satellite 1 in this sample network fails, and the groundstation operators know of the failure, the groundstations will communicate through Satellite 2 and Satellite 3. The change in the number of intermediate satellites influences the expected reliability of the path, and thus the unknown failure model. The known failure model, along with the routing algorithm, determines the number of intermediate satellites between groundstations and can influence the unknown failure model.

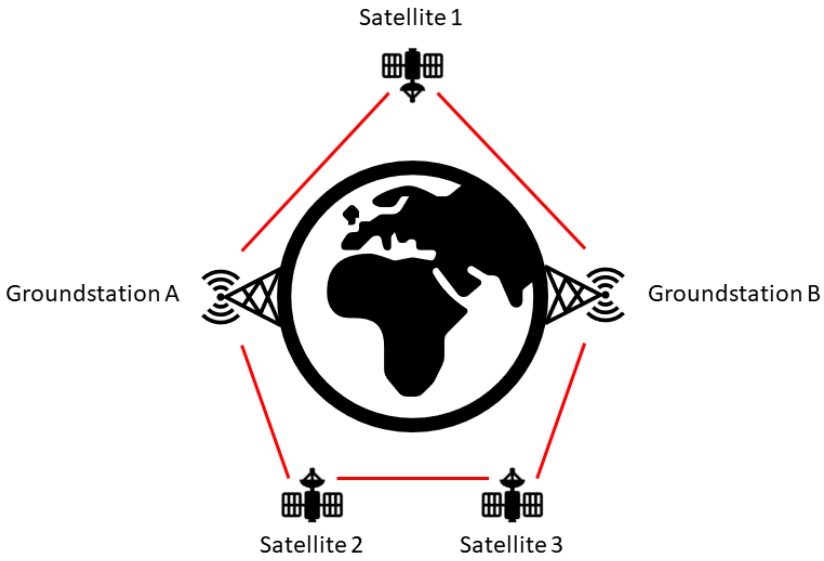

Figure 4.7: A sample network of three satellites with communication paths between two groundstations; groundstations A and B can communicate through Satellite 1, or through Satellite 2 and Satellite 3

To obtain a mean performance of the stochastic model, we want to model the most likely operational satellites known to the groundstation operators. By modeling all the possible constellation topologies, and the probability of the network transitioning to each of those, we can use Bayes' law to calculate the most likely network topology at any time and thus the most likely communication path between Groundstation A and B at time t. For the sample network in Figure 4.7, we can model the state space of the network topology of three satellites as eight possible states ($8 = 2^3$).

Using the state space of the sample topology in Figure 4.7 to create the eight states shown in Figure 4.8, we can create an 8x8 state transition matrix to describe the probability of transitioning from one state to another. This method can determine the likelihood of each topology at each time step, and can therefore yield a most likely estimate of the constellation failure model.

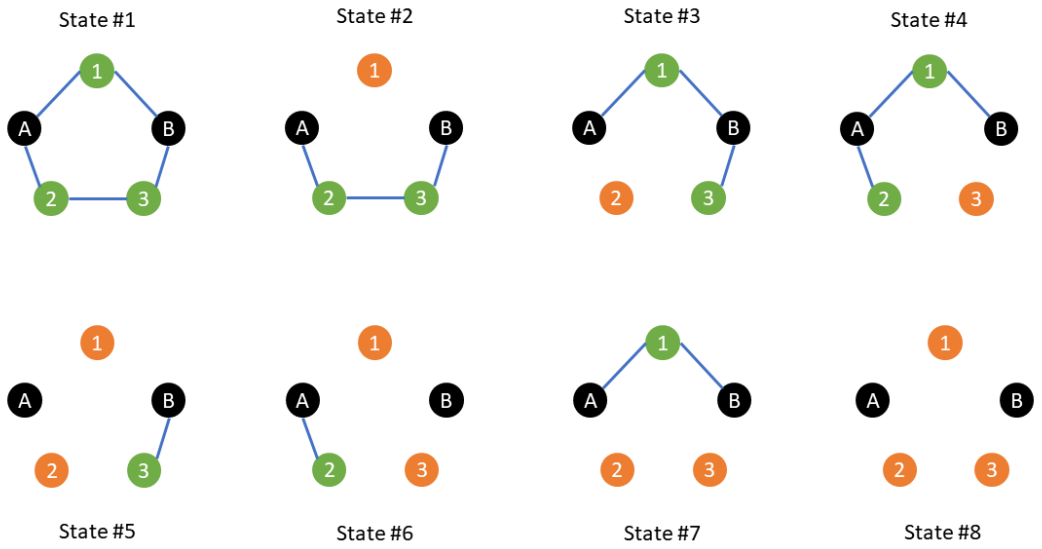

Figure 4.8: State space of the sample network topology in Figure 4.7, where black nodes are groundstations, green nodes are operational satellites, and orange nodes are failed satellites

While the state space transition method is feasible for small networks, it is not scalable to large satellite constellations. The number of states of the topology increases exponentially with respect to the number satellites in the constellation, making the state transition matrix extremely large and infeasible for simulating broadband satellite constellations of over 1,000 satellites. The second problem is that satellite failures are not the only way to change network topology; satellites

traveling through polar regions must disconnect satellite-to-satellite connections, due to the high relative velocity and difficulty in antenna pointing between satellites (see Section ***). This changes the number of states of the network between time steps, which makes updating the state-transition matrix more complex. For these reasons, we will not use a deterministic model for the known failure model.

To simulate known failures which change the network topology we will use a stochastic model of satellite failures at each time-step, using the satellite conditional reliability function, and run a Monte Carlo simulation to converge to a mean performance. Although computationally expensive, the Monte-Carlo approach allows us to model the groundstation operators' knowledge of failed satellites. Each time step, we update the network topology known to the operators based on the known failure model.

4.5 Communication Network Performance Metric Background

One of the most common performance metrics used to evaluate communication constellations is the availability of the system, which is the fraction of total uptime communication hours over some time period divided by the total time period length (Ayers, 2012). A point-to-point network's life-cycle availability, $A_{\text{PTP network}}$, can be calculated using the groundstation availability, $A_{\text{GS life cycle}}$, and the satellite availability, $A_{\text{satellite life cycle}}$, defined in Equations 4.7 to 4.9 (Ayers, 2012).

$$A_{\text{GS life cycle}} = \frac{\text{GS Uptime}}{\text{Total Time}} \tag{4.7}$$

$$A_{\text{satellite life cycle}} = \frac{\text{Satellite Uptime}}{\text{Total Time}} \tag{4.8}$$

$$A_{\text{PTP network}} = \left(A_{\text{GS life cycle}}\right)^2 \left(A_{\text{satellite life cycle}}\right) \tag{4.9}$$

The GS Uptime and Satellite Uptime are the median time to failure of each component, which is less than the total life cycle time for each system. This measure of network performance is a scalar value measuring the network performance over its lifetime. Network availability does not consider the volume of data throughput through the network at any given time.

4.6 Simulation Performance Metric

Ayers' network availability does not indicate the change in performance over time; for our model, we want to measure how resilient a constellation network is with component failures over time. We created a performance metric which estimates the network throughput at each timestep. We define $Q(t)$ as the rate of data sent from groundstations (bandwidth), and we define $H(t)$ as the expected data rate received by groundstations due to satellite failures. In a simple network with two groundstations with one-way communication links, the expected data received by a groundstation is less than the data sent, illustrated in Figure 4.9. Due to expected satellite failures, the expected data received by the receiving groundstation is less than the data sent by the transmitting groundstation.

$Q(t)$ = rate of data sent
$H(t)$ =expected throughput

$$H(t) \leq Q(t)$$

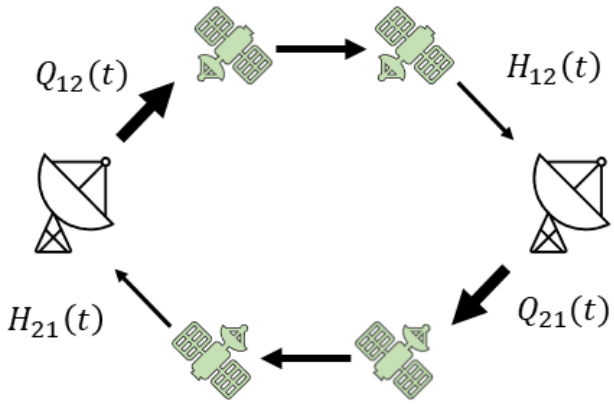

Figure 4.9: Sample constellation network with two identical groundstations sending one-way communication to each other through a series of satellites

We define a generic performance network, the total throughput fraction f_H, as a fraction of the total amount of data sent through the network divided by the total amount of data received by the network. For the network shown in Figure 4.9, the total throughput fraction is expressed in Equation (4.10).

$$f_H = \frac{H_{12} + H_{21}}{\mathrm{Q}_{12} + \mathrm{Q}_{21}} \tag{4.10}$$

The total throughput fraction is a measure of the overall network's performance in delivering data between all groundstations over time. Unlike the network availability measure, this value helps identify how quickly the network performance deteriorates over time with different satellite reliability functions. It also assumes that all groundstations are equally important, which is often the case in point-to-point networks. For a general constellation, where A_{ij} represents the data sent by groundstation i to groundstation j, and B_{ij} represents the data received by groundstation i from groundstation j, the general total throughput fraction can be expressed as a function of A_{ij} and B_{ij} for x groundstations:

$$f_H = \frac{\sum_{i=1}^{x}\sum_{i\neq j}^{x} A_{ij}}{\sum_{i=1}^{x}\sum_{i\neq j}^{x} B_{ij}} \tag{4.11}$$

Without any failures, and with fully connected groundstation nodes, $A_{ij} = B_{ji}$. When we include the two-part failure model, there is no analytical expression for B_{ji} in terms of A_{ij} because the number of known operational satellites between groundstations changes stochastically.

4.7 Summary

This chapter discusses the important features of our failure model which includes both known and unknown failures. We showed how we simulate unknown failure analytically using the expected functionality of satellites, and we simulate known failures using a Monte-Carlo simulation of stochastic failures. The total throughput fraction is a measure of the network's performance with failures, which will help distinguish the performance between satellite constellations with different satellite reliability functions.

5. RESULTS

5.1 Simulation Framework

We initialize our simulation with an initial topology of a Walker star constellation, where all satellites are fully functional. Using the satellite positions, we construct the network topology with the rule-based algorithm from Chapter 3 and determine the paths between groundstations using a shortest path algorithm. We then calculate the total throughput fraction of the network due to unknown failures over the next timestep. Following the performance metric calculation, we propagate the satellite orbits using the two-body model and simulate known failures at the next timestep. We then reconstruct the network topology using available operational satellites, and recompute the communication paths of the network. This iterative algorithm for our simulation is illustrated in Figure 5.1.

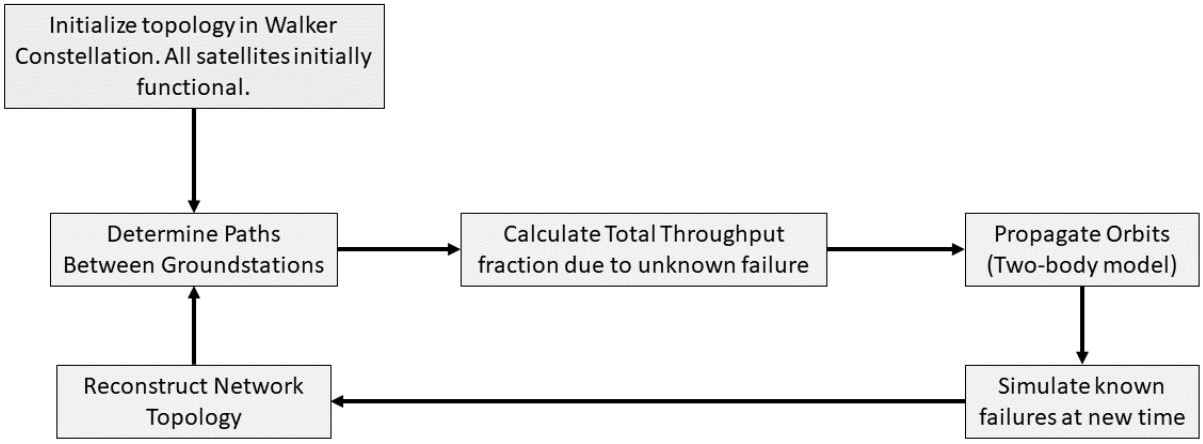

Figure 5.1: Block diagram of the simulation framework using all the major components presented in Chapters 2 to 4

5.2 Baseline Simulation Case

We created a baseline simulation case of a LEO broadband constellation with no satellite failures, and 1200 satellites equally distributed across 20 orbital planes in a Walker star constellation. Each circular orbital plane has an inclination of $i = 80^o$, and an altitude of 1200 km. Following the design of most LEO broadband constellations, we set the critical latitude at which cross-plane satellite communication links are disabled at $\pm 60^o$ latitude. We also use a

conservative minimum elevation angle of 40^o, which is the greatest elevation angle we found in the literature (del Portillo et al., 2019).

5.3 Timestep Analysis

The timestep we use in the simulation is a significant factor in the fidelity of our results. Although smaller timesteps yield higher-fidelity results, they are more computationally expensive. Our goal is to find the largest possible timestep without compromising accuracy. Each iteration of the simulation takes approximately 1 second and generates approximately 200 kb of data; over a simulation period of 3.7 years, we are computationally restricted by time and disk space. Restricting our total real simulation time to 14 days, and the disk space to 13.8 GB (maximum data size for processing in MATLAB), we consider the following time-step options with the corresponding real simulation time and total data size in Table 5.1**.**

Table 5.1: Simulation timestep with the corresponding real simulation time and data size

Simulation Time Step	Real Simulation Time	Data Size
2000s	16.2 hrs	11.67 Gb
3000s	10.8 hrs	7.78 Gb
5000s	6.48 hrs	4.67 Gb
7000s	4.63 hrs	3.33 Gb
10000s	3.24 hrs	2.33 Gb
15000s	2.16 hrs	1.56 Gb

Using the candidate time-steps in Table 5.1 we run the full simulation with a constant reliability of $R_k(t) = 1$, meaning that no satellites fail. We use the total network throughput performance metric of each of the simulations to compare the effect of time-step on the simulation accuracy. Note that these simulations are discrete approximations of the continuous behavior; we assume that the network performance is constant between time-steps. This assumption causes the difference in performance history with different timesteps. The simulation results in Figure 5.2 show that greater simulation timesteps produce a total throughput fraction which varies sinusoidally over time. The curves share a similar mean total throughput fraction, however the frequency of the curves decreases with greater timesteps. To compare the network performances with different timesteps more quantitatively, we will use the mean and natural frequency of the total throughput fraction for each simulation in in Figure 5.2.

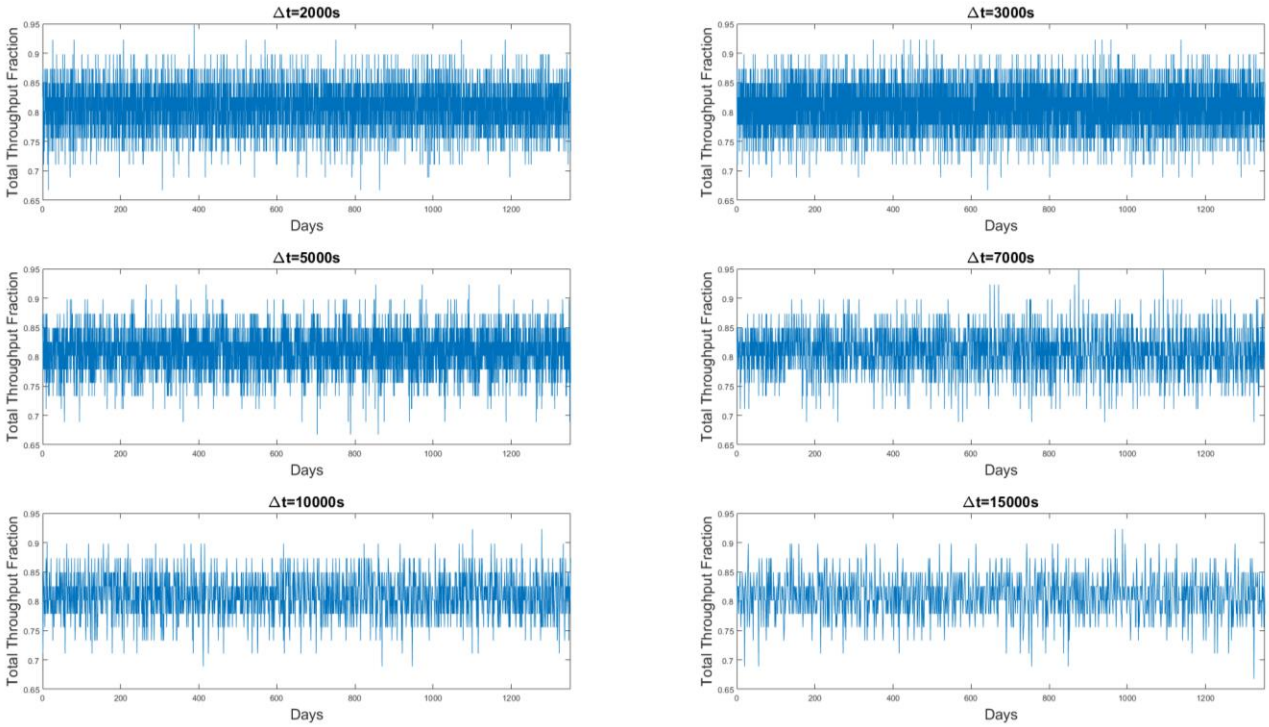

Figure 5.2: Total network throughput fraction for baseline LEO satellite constellation with no failures, using different timesteps (2000s, 3000s, 5000s, 7000s, 10000s, and 15000s)

The total throughput fraction within each simulation, shown in Figure 5.2, varies over time due to groundstations losing contact with satellites, and satellites near polar regions breaking their inter-planar satellite communication links. The mean total throughput fraction is a measure of the overall network's performance, independent of satellite failures. We expect that the mean total throughput fraction should be independent from the timestep used, over a significantly long period (greater than the period of the constellation). We also expect that the range of the total throughput fraction is greater at smaller timesteps, since they yield higher-fidelity results. The mean and the range of total throughput fraction for each of the simulations with different timesteps, shown in Table 5.2, points to a slight positive trend in the mean total throughput fraction, but the variation is not significant. Our hypothesis that the timestep does not change the overall network performance is supported by the small variation in the mean total throughput fraction.

Table 5.2: Mean total throughput fraction and range of the total throughput fraction for simulations of varying timesteps in Figure 5.2

Simulation Timestep	Mean Total Throughput Fraction	Range of Total Throughput Fraction
2000s	0.8090	0.2809
3000s	0.8090	0.2556
5000s	0.8092	0.2556
7000s	0.8092	0.2594
10000s	0.8095	0.2341
15000s	0.8096	0.2556

Table 5.2 shows that the range does not decrease significantly with increasing timesteps (3000 s - 15000 s) as expected; the range in fact increases from a timestep of 7000 s to a timestep of 10000 s. This analysis indicates that the total throughput performance range may not be strongly correlated to the simulation timestep.

To determine an appropriate simulation timestep for further analysis, we are interested in comparing the periodicity of the simulations with different timesteps to each other. A Fourier series is well-suited for approximating a periodic function with apparently irregular variations. By fitting a Fourier function to the data from each simulation, will reveal the natural frequency of the curves. We use an 8-series Fourier function for high fidelity, in the form:

$$\begin{aligned} f(x) = a_0 &+ a_1\cos(x\omega) + b_1\sin(x\omega) + a_2\cos(2x\omega) \\ &+ b_2\sin(2x\omega) + a_3\cos(3x\omega) + b_3\sin(3x\omega) \\ &+ a_4\cos(4x\omega) + b_4\sin(4x\omega) + a_5\cos(5x\omega) \\ &+ b_5\sin(5x\omega) + a_6\cos(6x\omega) + b_6\sin(6x\omega) \\ &+ a_7\cos(7x\omega) + b_7\sin(7x\omega) + a_8\cos(8x\omega) \\ &+ b_8\sin(8x\omega) \end{aligned} \tag{5.1}$$

where $a_i, b_i \in \mathbb{R}$ are coefficients of the Fourier function, and $\omega \in \mathbb{R}$ is the natural frequency. Fitting an 8-series Fourier function to each simulation curve from Figure 5.2, we can quantitatively compare the periodicity of the simulation results.

The Fourier series approximations of the datasets do not match the amplitude variation that the simulated datasets, shown in Figure 5.3, however we are more interested in the natural frequency of each approximation, ω. We also examine the constant coefficient a_0, which is closely related to the mean performance.

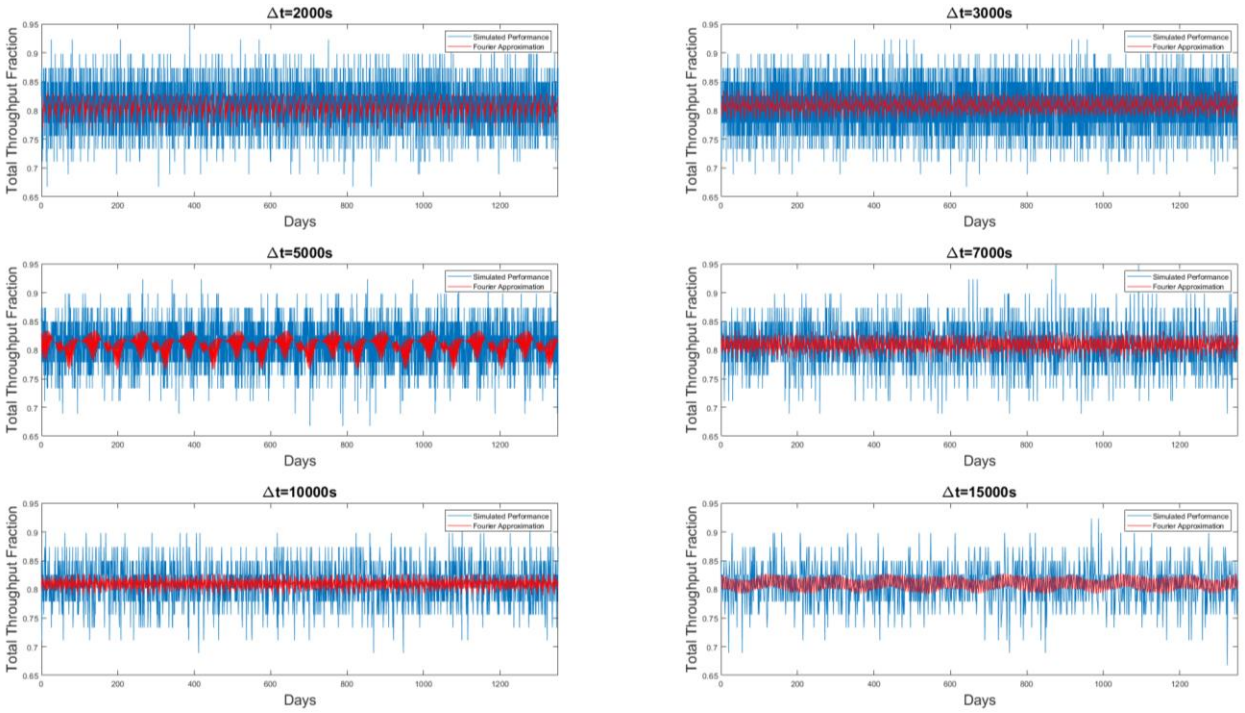

Figure 5.3: Total network throughput fraction for varying different timesteps (blue), with an 8-series Fourier approximation (red)

The frequency of the fitted 8-series Fourier function in Table 5.3 decreases with the simulation timestep. This is expected, since greater simulation timesteps means less data points and fidelity, and therefore a longer period and smaller frequency. The 95% confidence bounds of ω and the constant a_0 of the fitted Fourier functions is narrow, showing that this Fourier series is an accurate approximation of the simulation data. Plotting the natural frequencies of the fitted Fourier series in Figure 5.4, shows that there is a significant discontinuity in the natural frequency for timesteps greater than 7000s. Using this analysis as our basis, we selected the timestep $\Delta t = 7000\ s$ for our simulations.

Table 5.3: ω and a_0 parameters of the 8-series Fourier function fit to each dataset, with 95% confidence bounds

Simulation Timestep	**ω [1/s]**	**ω 95% Confidence Bounds (min, max)**	**a_0**	**a_0 95% Confidence Bounds (min, max)**
2000s	7.294E-5	(7.294E-5, 7.295E-5)	0.8090	(0.8082, 0.8098)
3000s	6.628E-5	(6.628E-5, 6.629E-5)	0.8090	(0.8082, 0.8098)
5000s	7.295E-5	(7.295E-5, 7.295E-5)	0.8092	(0.8081, 0.8104
7000s	6.597E-5	(6.596E-5, 6.597E-5)	0.8092	(0.8077, 0.8108)
10000s	1.331E-5	(1.331E-5, 1.331E-5)	0.8095	(0.8077, 0.8113)
15000s	2.023E-6	(2.021E-6, 2.024E-6)	0.8096	(0.8075, 0.8117)

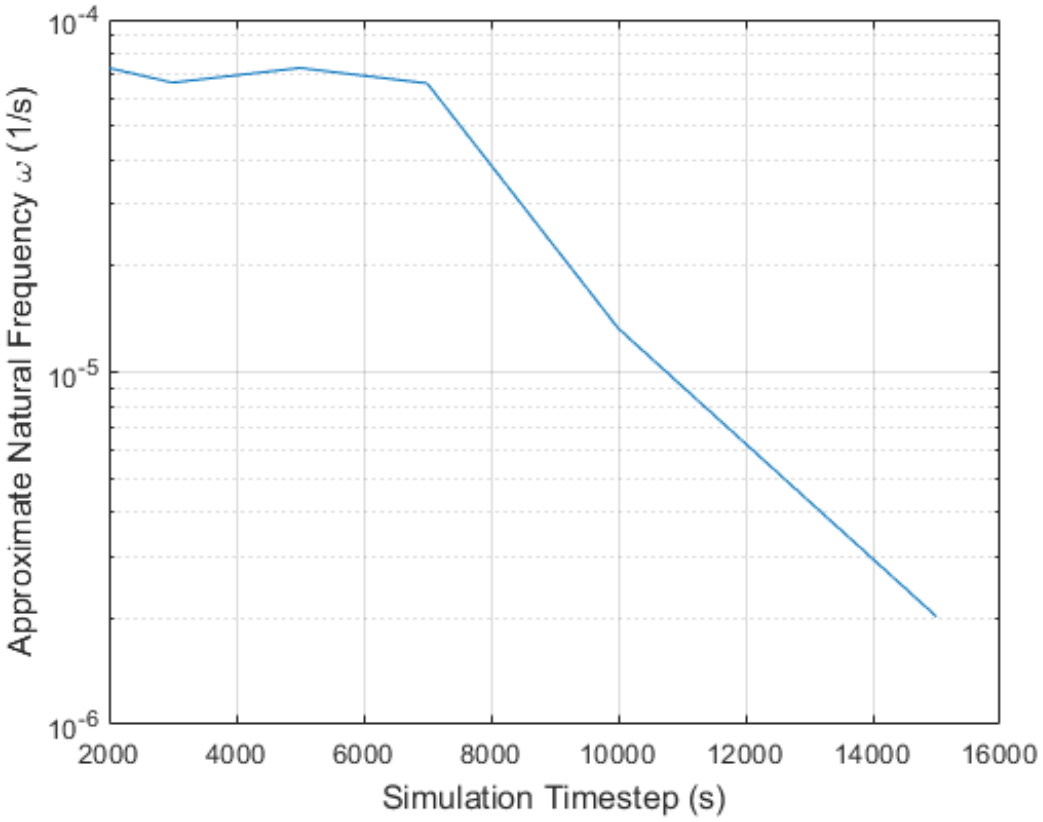

Figure 5.4: Natural frequency of fitted Fourier function of the total network throughput for simulations with varying timesteps

5.4 Failure Model Sensitivity

Although most LEO broadband satellites are less than 500 kg, which constitutes a small satellite, we want to test different reliability functions from other categories of satellites to see how sensitive the performance is to satellite reliability. We first test the constellation performance for a baseline case of perfectly reliable satellites that never fail. We then compare the results to constellation performance with satellite reliability functions of: MEO; Large-LEO; and Large-GEO satellites, where the fitted 2-Weibull reliability parameters of each function is shown in Table (Saleh and Castet, 2011). The reliability functions of each satellite category are shown in Figure 5.5.

Table 5.4: Parameters of the 2-Weibull Mixture function for each mass-orbit category of spacecraft (Saleh and Castet, 2011)

Satellite Category	$\boldsymbol{\alpha}$	$\boldsymbol{\beta_1}$	$\boldsymbol{\beta_2}$	$\boldsymbol{\theta_1}$	$\boldsymbol{\theta_1}$
Small-LEO	0.9759	0.0128	12.4386	1167882377.03	3.45
Medium	0.0128	1.0355	1.5336	0.20	103.08
Large-LEO	0.9559	0.7840	1.4070	65.82	1.04
Large-GEO	0.9057	0.4154	5.0600	46551.01	10.21

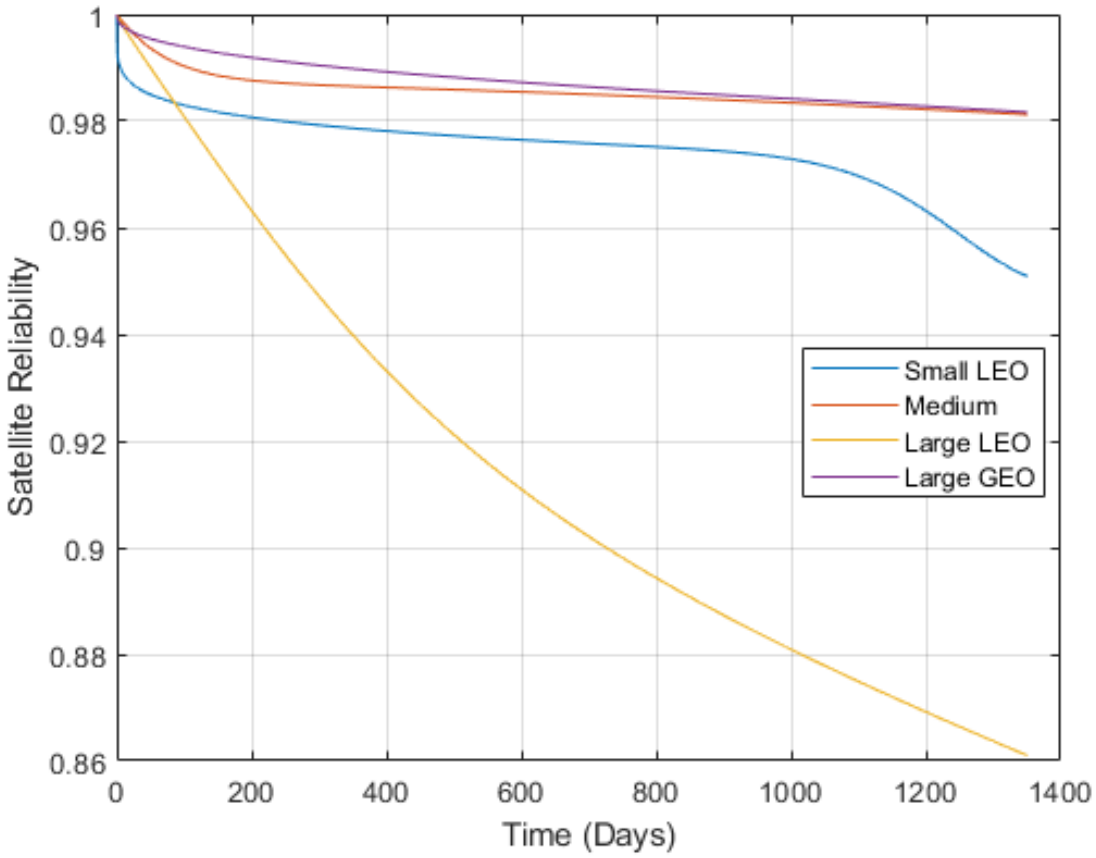

Figure 5.5: Reliability functions of each satellite category, represented as a 2-Weibull mixture function with the parameters in Table 5.4

Using the four reliability functions we ran four simulation cases over a time period of 3.7 years, averaging the total throughput fraction from 10 simulations for each case. We were limited to only 10 simulations for the Monte Carlo simulation due to the large data files that our simulation generates. The total throughput fraction quickly falls to zero within the first 100 days, shown in Figure 5.6. When we zoom in to the first 100 days of the simulation, Figure 5.7 shows that the Small-LEO satellite case fails significantly faster than the other three reliability cases. This makes sense, as the Small-LEO reliability is the smallest of the reliability functions in the first 100 days, shown in Figure 5.5, due to the infant mortality (Saleh and Castet, 2011). Two of the significant contributing factors to this are failures in the gyro subsystem, and failures in the thruster system, which are generally far less robust in small satellites compared to larger ones (Saleh and Castet, 2011).

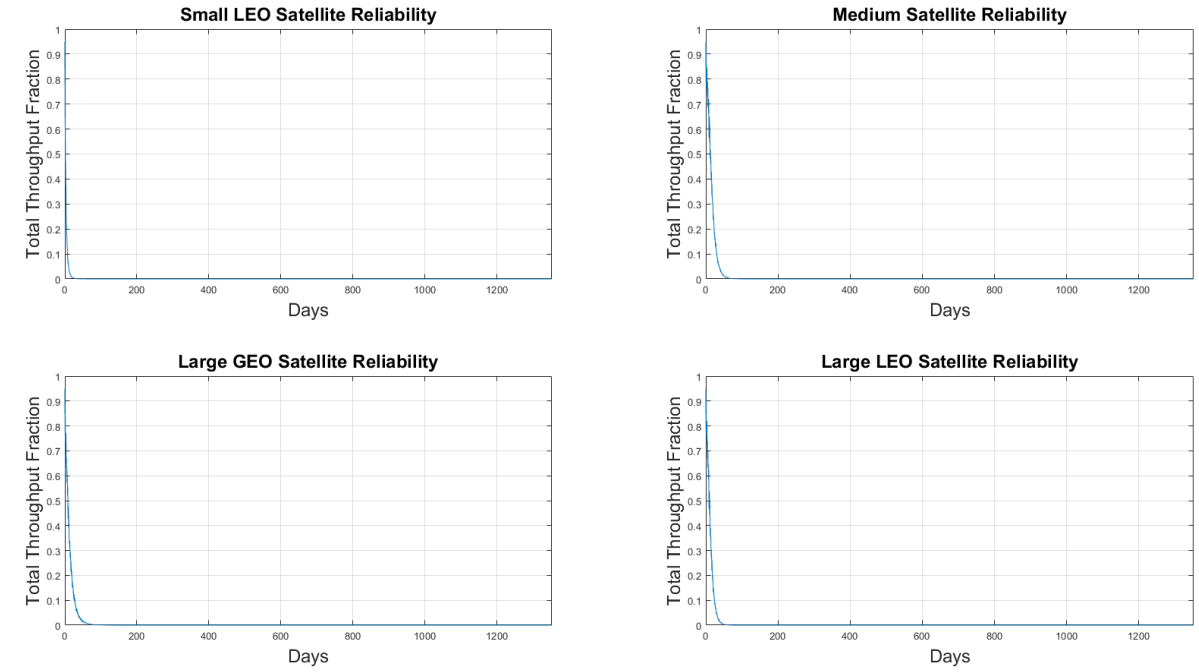

Figure 5.6: Total throughput fraction (mean of 10 runs per case) for four satellite reliability functions over 3.7 years

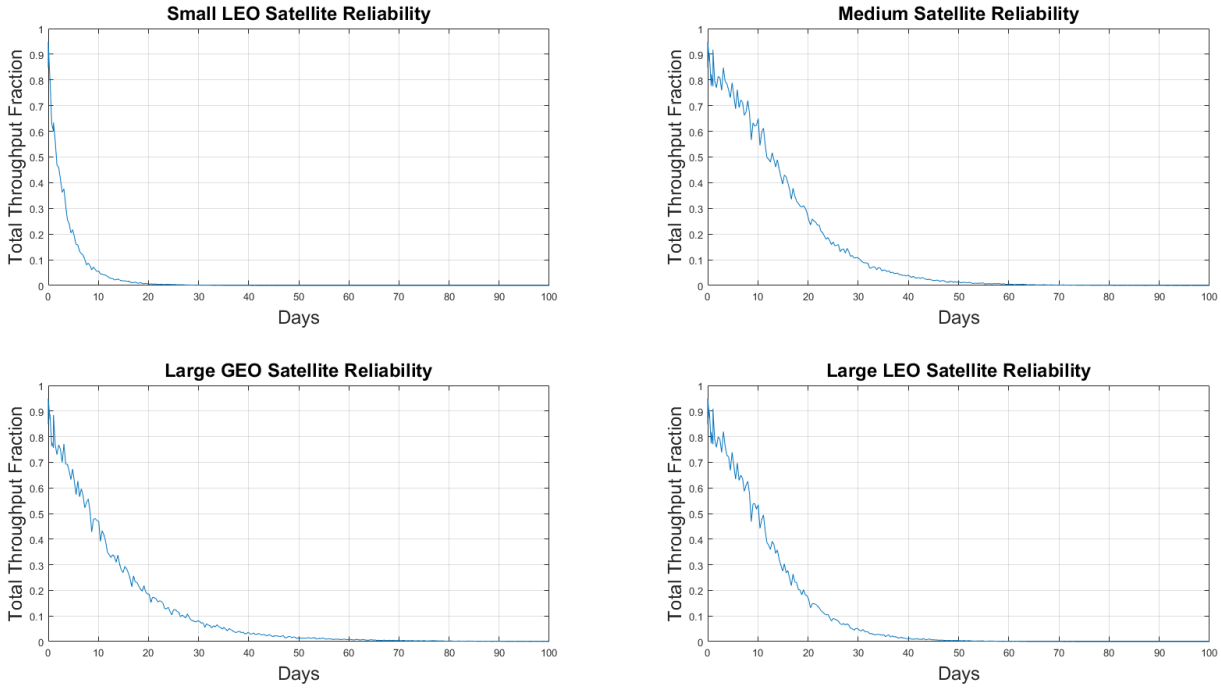

Figure 5.7: Total throughput fraction of four satellite reliability functions over 100 days

To observe the performance behavior over a longer time-period, thus diminishing the relative size of the timestep, we slow down the rate of deterioration by $\frac{1}{10000}$. Although we could have increased the constellation performance by using more satellites, we were limited by the maximum adjacency matrix size that Matlab can accommodate. Slowing down the rate of deterioration, the new satellite reliability functions are shown in Figure 5.8.

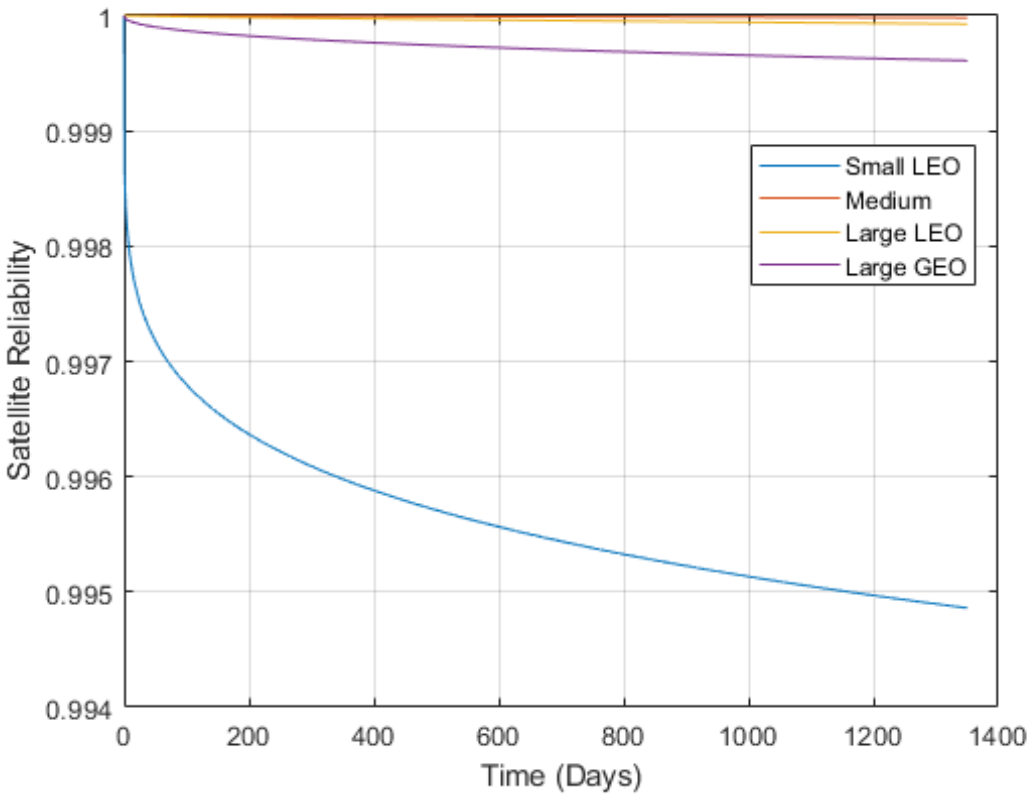

Figure 5.8: Modified satellite reliability functions, where the rate of deterioration is slowed by 1/10000 from the original reliability functions

Using the new reliability functions, the simulation results are much clearer over the 3.7 year simulation period. The Small LEO satellite reliability simulation still falls to zero very rapidly, however the cases with higher reliability do present some significant results, shown in Figure 5.9. The relative performance between the different simulation follows the same relative performance of each of the reliability functions; that is, the most reliable satellite type (Medium) has the highest total throughput fraction over time, while the least reliable satellite type (Small-LEO) has the lowest total throughput fraction compared to the other simulation cases.

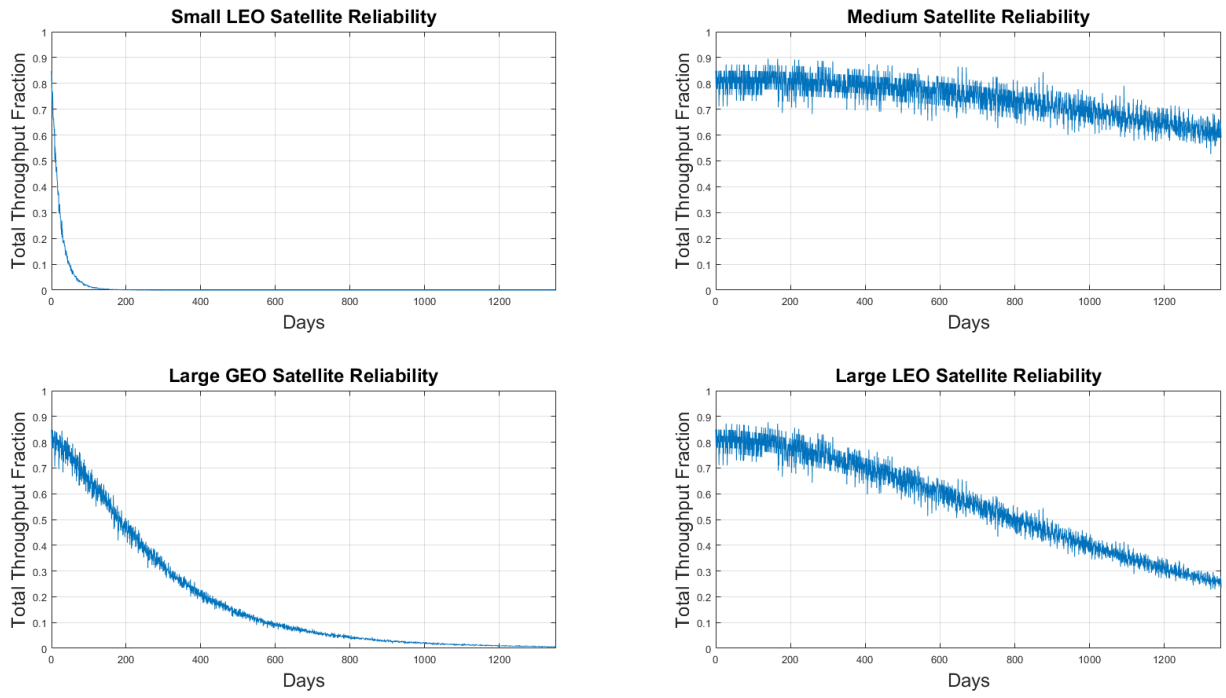

Figure 5.9: Total throughput fraction (mean of 10 runs per case) for the four modified satellite reliability functions, simulated over 3.7 years

5.5 Maximum Groundstation Link Sensitivity

We also conducted a sensitivity analysis testing our least confident assumption: the maximum number of groundstation-to-satellite links. Since there are currently no fully developed broadband constellation networks, we are using an average of 10 maximum groundstation-to-satellite links based on the number of large-dish antennas that are typically used at groundstations. We calculated the total throughput fraction of satellite constellations where the groundstation can have up to 5, 15, and 10 groundstation-to-satellite links for reliability functions with perfect reliability, modified small-LEO reliability, medium reliability, large-GEO reliability, and large-LEO reliability functions. The results of this sensitivity analysis in Figure 5.10 indicate that the maximum number of groundstation-to-satellite links does not have a significant impact on the network performance parameter. We found that this is because the average number of satellites in the field of view of groundstations is less than three. LEO satellite groundstations use a minimum of three or four antennas per groundstation (Ingram, et al., 2005), therefore testing a lower number of maximum groundstation links would not be useful.

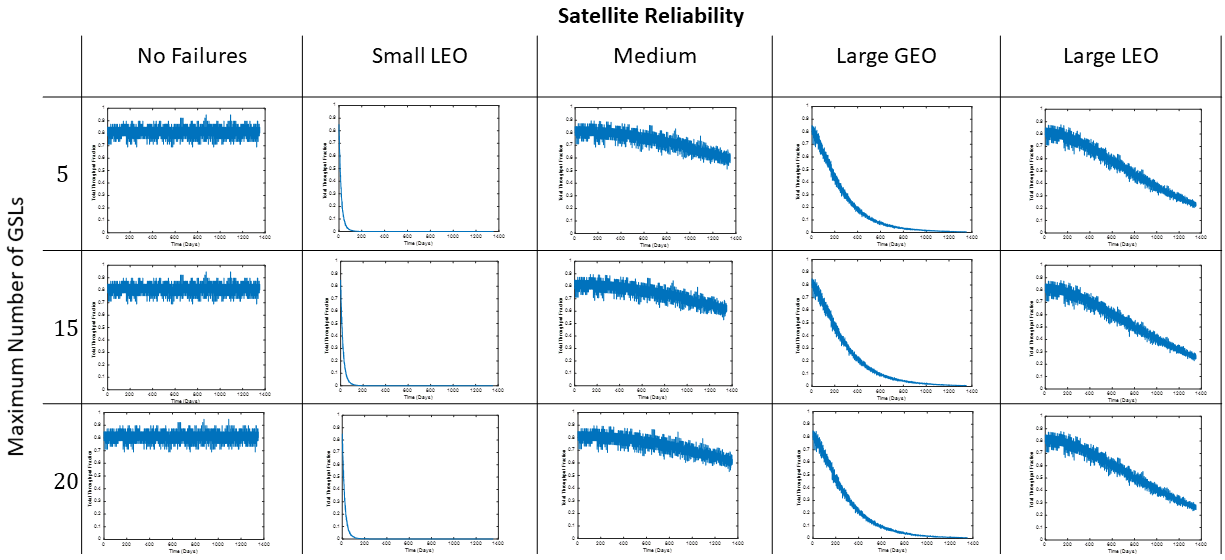

Figure 5.10: Total throughput fraction of three possible maximum number of groundstation-to-satellite links (GSLs), each with five different satellite reliability functions

5.6 Summary

In this chapter we defined a baseline constellation case, and tested the simulation with multiple timesteps. By comparing the natural frequencies of the performance metrics, we selected a timestep of 7000 s for the simulation model. We showed the constellation's performance for multiple satellite reliability functions, and also performed a sensitivity analysis of the number of groundstation-to-satellite links. We found that the number of satellite-to-groundstation links does not significantly affect the constellation network's performance for more than five groundstation-to-satellite links.

6. CONCLUSION

Using the simulation cases tested in this thesis, we can make several conclusions about the reliability of LEO broadband constellations. The first conclusion is that the reliability of satellites is too low for a functional constellation of less than 1,200 satellites. Particularly the small-LEO satellite reliability was too low to be operational for the LEO constellation that we tested. It is possible that satellites have become significantly more reliable, however we did not have a newer reliability function available.

The second major conclusion is that the maximum number of groundstation-to-satellite links is not a significant factor in improving the constellation performance. Because of the relatively small constellation we tested compared to the number of groundstations, the satellites were too sparsely spaced.

Lastly, we conclude that the simulation timestep is a significant factor in improving the fidelity of this model. We found that from a set of candidate simulation timesteps, 7000 s is an appropriate timestep given our computational limitations.

6.1 Lessons Learned

Through this work, we learned how to construct a large simulation model to test the reliability of satellites in a communication constellation. Most significantly, we designed a new method for simulating satellite failures—through known failure, and unknown failure models.

We also learned that computational limitations are a driving factor in testing our model. We used MATLAB due to its advantages in computing graph properties, however it is not suitable for large data computations.

6.2 Future Work

In future work, several areas of this simulation can be examined more closely. The first area would be in the timestep analysis. Using our method for determining an appropriate timestep, it would only work for our baseline constellation model. For testing different constellation models, one would need to recompute an appropriate timestep according to their computational limitations.

Another area of future work would be in including partial errors in satellite communication links, and satellite subsystems. We assumed that satellite communication links are fully reliable, however in inclement weather satellite communication links are often only partially operational. The satellite subsystems can also be an area of improvement of this work, however it would require much more computational power.

APPENDIX A. CANDIDATE GROUNDSTATIONS

Groundstation positions and country information for constellation ground segment (del Portillo et al., 2019).

Table 6.1: Groundstations and country information for groundstation ID 1 to 40

ID	Name	Code	Latitude	Longitude	Country	Organization	Region
1	Alaska Satellite Facility	ASF	64.86	-147.85	USA	NEN (NASA)	N. America
2	Clewiston	CLE	26.73	-82.03	USA	SSC	N. America
3	Esrange	ESR	67.88	21.07	Sweden	SSC	Europe
4	Florida Ground Station	FGS	29.00	-81.00	USA	NEN	(NASA) N. America
5	Fucino	FUC	42.00	13.55	Italy	SSC	Europe
6	Hartebeesthoek	HBK	-25.64	28.08	South Africa	SSC	Africa
7	Inuvik	INU	68.40	-133.50	Canada	SSC	N. America
8	McMurdo Ground Station	MMGS	-77.81	166.69	Antartica	NEN	(NASA) Oceania
9	O'Higgins	O'H	-63.32	-57.90	Antartica	SSC	S. America
10	Punta Arenas	PAN	-53.00	-71.00	Argentina	SSC	S. America
11	Santiago Satellite Station	SSS	-33.13	-70.67	Chile	SSC	S. America
12	Svalbard Ground Station	SGS	78.22	15.39	Norway	NEN	(NASA) Europe
13	USN Western Australia	USNWA	-29.05	114.90	Australia	SSC	Oceania
14	Wallops Flight Facility	WFF	37.94	-75.49	USA	NEN	(NASA) N. America
15	Weilheim	WEIL	47.84	11.14	Germany	SSC	Europe
16	Hawaii	HAW	19.82	-155.47	USA	KSAT	N. America
17	Tokyo	TOK	35.69	139.69	Japan	KSAT	Asia
18	Singapore	SIA	1.35	103.82	Singapore	KSAT	Asia
19	Trollsat	TROLL	-72.10	2.32	Antarctica	KSAT	Antarctica
20	Vardo	VARD	70.37	31.10	Norway	KSAT	Europe
21	Tromso	TROM	69.65	18.96	Norway	KSAT	Europe
22	Grimstad	GRIM	58.34	8.59	Norway	KSAT	Europe
23	Puertollano	PTLL	38.69	-4.11	Spain	KSAT	Europe
24	Dubai	DUB	25.2	55.27	UAE	KSAT	Asia
25	Mauritius	MAUR	-20.35	57.55	Mauritius	KSAT	Africa
26	Panama	PNM	8.54	-80.78	Panama	KSAT	S. America
27	Central Africa	AFR	4.84	10.1	Central Africa	KSAT Africa	Africa
28	New Zealand	NZL	-46.02	167.81	New Zealand	KSAT Oceania	KSAT
29	Kourou	KOU	5.16	-52.65	French Guiana	ESA	S. America
30	Redu	REDU	50	5.16	Belgium	ESA	Europe
31	Cebreros	CBRR	40.46	4.46	Spain	ESA	Europe
32	Villafranca	VILLA	40.26	-3.57	Spain	ESA	Europe
33	Maspalomas	MSPL	27.45	-15.38	Spain	ESA	Europe
34	Santa Maria	STMAR	36.59	-25.08	Portugal	ESA	Europe
35	Malargue	MLG	-25.78	-69.4	Argentina	ESA	S. America
36	Frankfurt	FRKT	50.12	9.92	Germany	ESA	Europe
37	Perth	PERT	-31.8	115.89	Australia	ESA	Oceania
38	Delhi	DELH	28.55	77.29	India	Viasat	Asia
39	Chennai	CHNN	13.13	80.17	India	Viasat	Asia
40	Chengdu	CHND	30.58	104.11	China	Other	Asia

Table 6.2: Groundstations and country information for groundstation ID 41 to 77

41	Jakarta	JKRT	-6.34	106.86	Indonesia	Indosat	Asia
42	Novosibirk	NOVO	55.02	82.84	Russia	Other	Asia
43	Rio de Janeiro	RIO	-22.98	-43.35	Brazil	SES	S. America
44	Addis Ababa	ADDIS	9.01	38.76	Ethiopia	SES	Africa
45	New Caledonia	NEWC	-22.26	166.4	New Caledonia	Intelsat	Oceania
46	Guam	GUAM	13.42	144.75	USA	NEN (NASA)	Asia
47	Saint Helena	STHEL	-15.97	-5.71	Saint Helena	Other	Africa
48	Sychelles	SYCH	-4.63	55.45	Sychelles	Laban	Africa
49	Luanda	LUAND	-8.84	13.23	Angola	Other	Africa
50	Barbados	BARB	13.1	-59.63	Barbados	Other	S. America
51	Alaska	HOME	59.65	-151.54	USA	NEN (NASA)	N. America
52	Nuuk	NUUK	64.18	-51.74	Greenland	Other	N. America
53	Sapporo	SAPP	43.06	141.34	Japan	JAXA	Asia
54	Adelaide	ADEL	-34.93	138.6	Australia	SES	Oceania
55	Accra	ACCR	5.56	-0.2	Gahna	SES	Africa
56	Lagos	LAGS	6.52	3.38	Nigeria	SES	Africa
57	Lurin	LRIN	-12.25	-76.88	Peru	SES	S. America
58	Hortolandia	HORTO	-22.85	-47.21	Brazil	SES	S. America
59	Dijbouti	DJIBO	11.83	42.59		SES	Africa
60	Abu Dhabi	ABUDH	24.45	54.38	UAE	SES	Asia
61	Kowoloon	KWLO	22.32	114.18	Hong Kong	SES	Asia
62	Brewster	BREW	48.09	-119.78	USA	SES	N. America
63	Los Angeles	LA	34.05	-118.24	USA	SES	N. America
64	Vernon	VERN	34.15	-99.27	USA	SES	N. America
65	Karachi	KRCH	24.86	67.1	Pakistan	SES	Asia
66	Kiev	KIEV	50.45	30.52	Ukraine	SES	Europe
67	Dubbo	DBBO	-32.23	148.63	Australia	SES	Oceania
68	Denver	DENV	39.74	-104.99	USA	Intelsat	N. America
69	Kumsan	KUMS	35.36	128.41	South Korea	Intelsat	Asia
70	Napa	NAPA	38.25	-122.28	USA	Intelsat	N. America
71	Ottawa	OTTA	45.42	-75.7	Canada	Telesat	N. America
72	Yellowknife	YLWKN	62.45	-114.37	Canada	Telesat	N. America
73	St. John’s	STJHN	47.56	-52.71	Canada	Telesat	N. America
74	Iqaluit	IQLT	63.75	-68.52	Canada	Telesat	N. America
75	Saskatoon	SSKAT	52.13	-106.67	Canada	Telesat	N. America
76	Mexico DF	MEXDF	19.43	-99.13	Mexico	Eutelsat	N. America
77	Cape Verde	CAPE	14.55	-23.31	Cape Verde	Other	Africa

APPENDIX B. MATLAB CODE

```
% Islam Nazmy
% This script propagates the satellite broadband constellation defined and
% calculates the NPV at each time step, based on the failure model and the
% network model.

clc; close all; clear all;

%% Initialize the Simulation Clock
t0 = datetime('2020-06-12 20:00:00'); % [date] starting epoch
dt = 10000; % [seconds]
t_sim = years(3.7);
sim_clock = SimClock(t0,dt,t_sim);
ctr = 1;

%% Initialize the Satellite Constellation at epoch_0
const_n = 60; % number of satellites per orbit
const_m = 20; % number of orbital planes
const_ecc = zeros(1,const_m); % eccentricity of orbits
const_inc = deg2rad(80)*ones(1,const_m); % [rad] inclination of orbits
const_alt_peri = 1500*ones(1,const_m); % [km] perigee altitude of orbits
const_RAAN_0 = zeros(1,const_m);
const_arg_peri_0 = zeros(1,const_m);
const_TA_0 = 0;
J2 = 0; %1.08262668E-3;
GS_band = 1; % [Gbps] data bandwidth to be delivered from GS to each of
the other GS's

satellite_constellation = SatelliteConstellation(const_n, const_m,
const_ecc, const_inc, const_alt_peri, const_RAAN_0, const_arg_peri_0,
const_TA_0, J2);


%% Initialize the Groundstations
% fname = 'groundstations\IXP_locations_77_Portillo_raw.txt';
% [fid,msg] = fopen(fname);
```

```
    % txt_cell = textscan(fid,'%q');
    % txt = txt_cell{1};
    % fclose(fid);
    %
    % name_gs=[]; code_gs=[]; lat_gs=[]; long_gs=[];
    % for i = 0:8:(length(txt)-8) % 77 in total groundstations in file
    %     name_gs = cat(1,name_gs,txt(i+2));
    %     code_gs = cat(1,code_gs,txt(i+3));
    %     lat_gs = cat(1,lat_gs,str2double(txt{i+4}));
    %     long_gs = cat(1,long_gs,str2double(txt{i+5}));
    % end
    % num_of_gs = length(name_gs); % total number of groundstations
    load('gs_portillo_data')

    ground_stations = GroundStations(name_gs, code_gs, lat_gs, long_gs,
sim_clock);


    %% Define the Failure Model
    satellite_type = "No_Failures";
    failure_model = Failure_Model(satellite_constellation, ground_stations,
satellite_type);

    %% Create the constellation network (satellites and groundstations)
    lat_limit = 60; % [deg] set latitude limit of satellite for inter-satellite
links. Greater than this limit, satellites will not make inter-planar
connections (connections to satellites in different planes)
    gs_min_elev = 40; % [deg] elevation angle of groundstations
    max_GSL = 10; % maximum number of groundstation-satellite links for a
single groundstation
    lat_limit = deg2rad(lat_limit); gs_min_elev = deg2rad(gs_min_elev);

    constellation_network = Constellation_Network(failure_model.sats_func,
failure_model.gs_func, satellite_constellation, ground_stations, lat_limit,
gs_min_elev, max_GSL, GS_band);

    %% Routing Algorithm
    routing_algorithm = ShortestPath_Routing_Algorithm(constellation_network);
```

```
    %% Analyze Throughput
    network_performance = Performance_Analysis(failure_model,
constellation_network.GS_to_GS_band, routing_algorithm);

    %% Save data file
    fname_save =
sprintf('Simulation_Data/%s/ctr_%d_dt_%d_FailureFunction_%s_DateTime_%s.mat',
test_name,ctr,dt,satellite_type,datestr(now,'ddmmyyyy_HHMMSS'));
    save(fname_save,'sim_clock','constellation_network','network_performance'
,'-v7.3')

    %% Run the simulation
    while sim_clock.go
        ctr = ctr + 1;
        tic

    % sat_pos_plot =
reshape([satellite_constellation.sat_rv{:}],[6,const_n*const_m])';
    %     node_pos = [sat_pos_plot(:,1:3); ground_stations.gs_pos];
    %     G = graph(constellation_network.adj_matrix,'upper');
    %     figure(1)
    %     clf
    %     hold on
    % Network_plot_3D =
plot(G,'XData',node_pos(:,1),'YData',node_pos(:,2),'ZData',node_pos(:,3),'Nod
eColor','r','MarkerSize',2,'EdgeColor',[0, 0.4470, 0.7410]);
    %
highlight(Network_plot_3D,(const_n*const_m+1):(const_n*const_m+num_of_gs),'No
deColor','y','MarkerSize',5)
    %     globeAxis = axesm('globe','Geoid',[6378 0]);
    %    gridLines = gridm('GLineStyle','-','Gcolor',[.7 .8 .9],'Grid','on');
    %     set(gca,'Box','off','Projection','perspective')
    %     load topo
    %     Earth = geoshow(topo,topolegend,'DisplayType','texturemap');
    %     demcmap(topo)
    %     rotate(gridLines,[0,0,1],rad2deg(gstime(sim_clock.time)))
    %     rotate(Earth,[0,0,1],rad2deg(gstime(sim_clock.time)))
```

```
    %     set(gcf, 'Position',  [100, 100, 900, 900],'color','k')
    %     set(gca,'visible','off')
    %     hold off
    %     axis equal
    %     view(0,30)
    %     pause(0.2)

    %     figure(2)
    %     clf
    %     geoshow('landareas.shp','FaceColor',[0.5 1 0.5]);
    %     hold on
    %                                          plot(ground_stations.gs_long,
ground_stations.gs_lat,'.k','MarkerSize',60);
    %    sat_pos_projected = sat_pos_plot(:,1:3)*ground_stations.rot_transf';
    %                                [sat_long,      sat_lat,      ~]      =
cart2sph(sat_pos_projected(:,1),sat_pos_projected(:,2),sat_pos_projected(:,3)
);
    %     nodes_long = [rad2deg(sat_long); ground_stations.gs_long];
    %     nodes_lat = [rad2deg(sat_lat); ground_stations.gs_lat];
    %                                                Network_plot_2D         =
plot(G,'XData',nodes_long,'YData',nodes_lat,'NodeColor','r','MarkerSize',1,'E
dgeColor',[0, 0.4470, 0.7410]);
    %
highlight(Network_plot_2D,(const_n*const_m+1):(const_n*const_m+num_of_gs),'No
deColor','k','MarkerSize',5)
    %     plot(rad2deg(sat_long),rad2deg(sat_lat),'r.','MarkerSize',30);
    %     axis([-70, -42, 55, 70])
    %     hold off

        sim_clock.stepTime();
        satellite_constellation                                              =
satellite_constellation.stepSatellites(sim_clock);
        ground_stations = ground_stations.stepGroundStations();
        failure_model = failure_model.updateFunctionality(sim_clock);
        constellation_network                                                =
constellation_network.updateAdjMatrix(failure_model.sats_func,
failure_model.gs_func, satellite_constellation.sat_rv, ground_stations.gs_pos);
```

```
        routing_algorithm
=routing_algorithm.getRoutingPaths(constellation_network.adj_matrix);
        network_performance                                                    =
network_performance.getNodeThru(failure_model.exp_ft,
constellation_network.GS_to_GS_band, routing_algorithm.routing_paths);

        fname_save                                                             =
sprintf('Simulation_Data/test_1/ctr_%d_dt_%d_FailureFunction_%s_DateTime_%s.m
at',ctr,dt,satellite_type,datestr(now,'ddmmyyyy_HHMMSS'));

save(fname_save,'sim_clock','constellation_network','network_performance','-
v7.3')

        toc
    end

    % SatelliteConstellation.m
    % This classdef defines a satellite Walker constellation with m orbital
    % planes and n satellites in each orbit. Each of the orbits are equally
    % spaced by Omega (RAAN), and each satellite in the orbits are equally
    % spaced by True Anomaly.
    % Create by Islam Nazmy
    % Last Modified 05/17/20
    classdef SatelliteConstellation
        properties
            n
                % number of satellites per orbit
            m
                % number of orbital planes
            ecc0
                % 1xm row array eccentricity of orbits
            inc0
                % [rad] inclination of orbits (single value, all the same)
            hp0
                % [km] perigee altitude of orbits (single value, all the same)
            Omega0
                % [rad] right ascension of the ascending node of the first
                % plane in the constellation. Assume all equally spaced.
```

```
            omega0
                % [rad] argument of the perigee for each plane in the
                % constellation
            Theta0
                % [rad] Initial true anomaly of the first satellite in the
                % plane. Assume all equally spaced satellites.
            rE = 6371;
                % [km] mean radius of Earth
            mu = 3.986E5;
                % [km3/s2] Earth gravitational constant
            J2
                % J2 constant for Earth spherical harmonics
            sat_rv
                % nxm cell matrix of satellites' position and velocity (in  the
                % geocentric equatorial system), calculated using Vallado's
                % coe2rv function from classical orbital elements. The entry
of
                % each cell is [r_i r_j r_k v_i v_j v_k]
        end
        methods
            % initialization constructor
            function obj = SatelliteConstellation(n_sats_per_orbit, m_planes,
eccentricity, inclination, alt_peri, RAAN_0, arg_peri_0, TA_0, J2_harmonics)
                if nargin ~= 0
                    obj.n = n_sats_per_orbit;
                    obj.m = m_planes;
                    obj.ecc0 = eccentricity;
                    obj.inc0 = inclination;
                    obj.hp0 = alt_peri;
                    obj.Omega0 = RAAN_0;
                    obj.omega0 = arg_peri_0;
                    obj.Theta0 = TA_0;
                    obj.J2 = J2_harmonics;

                    obj.sat_rv = cell(obj.n,obj.m);

                    rp = obj.hp0 + obj.rE;
                    sma = rp./(1-obj.ecc0);
```

```
        %             slr = sma.*(1-obj.ecc0.^2);

                    % calculate satellite states of Walker Constellation
                    mean_motion = sqrt(obj.mu./sma.^3); % [rad/s^2] mean
motion of each orbit (row array)
                    ang_mom = mean_motion.*sma.*(sma.*sqrt(1-obj.ecc0.^2)); %
angular momentum of each orbit (row array)

                    D_RA = linspace(0,pi,obj.m); % only use half sphere since
we do not want orbits overlapping
                    RA = obj.Omega0 + D_RA; % right ascension of each orbit

                    D_TA = 2*pi/obj.n; % change in true anomaly between
satellites in a plane
                    for j = 1:obj.m
                        for i = 1:obj.n
                            p = ang_mom(j)^2/obj.mu;
                            ecc = obj.ecc0(j);
                            incl = obj.inc0(j);
                            argp = obj.omega0(j);
                            TA = obj.Theta0 + (i-1)*D_TA; % true anomaly of
orbits
                            arglat = TA + obj.omega0(j);
                            truelon = arglat + RA(j);
                            lonper = RA(j) + obj.omega0(j);

                            obj.sat_rv{i,j}                                =
coe2rv(p,ecc,incl,RA(j),argp,TA,arglat,truelon,lonper);
                        end
                    end
                end
            end

            % propagate COEs using two-body model
            function [obj] = stepSatellites(obj,sim_clock)
                dt = sim_clock.increment;
                for j = 1:obj.m
                    for i = 1:obj.n
```

```
                    ri = obj.sat_rv{i,j}(1:3);
                    vi = obj.sat_rv{i,j}(4:6);
                    [rf, vf] = twobody3(obj.mu, dt, ri, vi);
                    obj.sat_rv{i,j} = [rf, vf];
                end
            end
        end

    %         % Gauss' variational equations (all angles in radians)
    %         function [dfdt] = rates(obj,t,f)
    %             % Orbital elements at time t
    %             h = f(1);
    %             ecc = f(2);
    %             RA = f(3);
    %             inc = f(4);
    %             w = f(5);
    %             TA = f(6);
    %
    %             r = h^2/obj.mu/(1+ecc*cos(TA)); % radius
    %             u = w + TA; % argument of latitude
    %
    %             % Orbital element rates at time t (Pg. 147 in
    %                                                                    %
https://booksite.elsevier.com/9780080977478/downloads/Appendix_D.pdf)
    %             hdot = -3/2*obj.J2*obj.mu*obj.rE^2/r^3*sin(inc)^2*sin(2*u);
    %                                                                edot     =
3/2*obj.J2*obj.mu*obj.rE^2/h/r^3*(h^2/obj.mu/r*sin(TA)*(3*sin(inc)^2*sin(u)^2
-1)-sin(2*u)*sin(inc)^2*((2+ecc*cos(TA))*cos(TA)+ecc));
    %                                                               TAdot     =
h/r^2+3/2*obj.J2*obj.mu*obj.rE^2/ecc/h/r^3*(h^2/obj.mu/r*cos(TA)*(3*sin(inc)^
2*sin(u)^2-1)+sin(2*u)*sin(inc)^2*sin(TA)*(h^2/obj.mu/r+1));
    %             RAdot = -3*obj.J2*obj.mu*obj.rE^2/h/r^3*sin(u)^2*cos(inc);
    %            idot = -3/4*obj.J2*obj.mu*obj.rE^2/h/r^3*sin(2*u)*sin(2*inc);
    %                          wdot  =  3/2*obj.J2*obj.mu*obj.rE^2/ecc/h/r^3*(-
h^2/obj.mu/r*cos(TA)*(3*sin(inc)^2*sin(u)^2-1)-
sin(2*u)*sin(inc)^2*sin(TA)*(2+ecc*cos(TA))+2*ecc*cos(inc)^2*sin(u)^2);
    %
    %             dfdt = [hdot edot RAdot idot wdot TAdot]';
```

```
%          end
    end
end
```

```
% GroundStations.m
% This classdef defines the groundstations for the constellation network.
% This class holds and updates the position vectors of each of the
% groundstations in the inertial frame, using lat/long as inputs.

classdef GroundStations
    properties
        name
            % full name of the groundstations
        name_code
            % code abbreviation of the groundstation names
        gs_lat
            % latitude of the groundstations [deg]
        gs_long
            % longitude of the groundstations [deg]
        gs_pos
            % groundstation position vectors [x y z] in the inertial frame
        rot_rate = 7.29212352E-5;
            % [rad/s] rotation rate of the Earth
        rot_transf
            % rotation transformation matrix due to Earth rotation
    end
    methods
        % initialization constructor
        function obj = GroundStations(gs_name, gs_name_code, gs_lat,
gs_long, sim_clock)
            if nargin ~= 0
                obj.name = gs_name;
                obj.name_code = gs_name_code;
                obj.gs_lat = gs_lat;
                obj.gs_long = gs_long;

                re = 6378.137;          % km
                gs_pos_x      =      re*cosd(gs_lat).*cosd(gs_long      +
rad2deg(gstime(sim_clock.time)));
                gs_pos_y      =      re*cosd(gs_lat).*sind(gs_long      +
rad2deg(gstime(sim_clock.time)));
                gs_pos_z = re*sind(gs_lat);
```

```
                    obj.gs_pos = [gs_pos_x, gs_pos_y, gs_pos_z]; % cartesian
coordinates of groundstations in form [x y z]

                    dt = sim_clock.increment; % increment (in seconds)
                    obj.rot_transf      =      [cos(-obj.rot_rate*dt),-sin(-
obj.rot_rate*dt), 0; sin(-obj.rot_rate*dt), cos(-obj.rot_rate*dt), 0; 0, 0,
1]; % [rad/s] rotation matrix of the Earth
                end
            end
            % rotate groundstations position vectors as Earth rotates
            function [obj] = stepGroundStations(obj)
                obj.gs_pos = obj.gs_pos*obj.rot_transf;
            end
        end
    end
```

```
    % Failure_Model.m
    % This clasdef defines the failure model of the satellite constellation,
as
    % a function of a single satellite, and the number of satellites in a
    % routing path.

    classdef Failure_Model
        properties
            sats_func
                % functionality of the satellites in the network, in the same
                % format as the cell matrix in Satellite_Constellation. This
                % nxm matrix (for m planes, with n satellites in each plane)
                % indicates 0 for failed satellites, and 1 for operational
                % satellites.
            gs_func
                % functionality of the groundstations. 1 indicates working, 0
                % indicates not operational
            exp_ft
                % expected functionality array
            sat_type
                % satellite type: No_Failures; Small_LEO; Medium; Large_LEO;
                % Large_GEO
        end
        methods
            % initialization constructor
            function     obj     =     Failure_Model(satellite_constellation,
ground_stations, sat_failure_type)
                obj.sats_func = true(size(satellite_constellation.sat_rv)); %
initialize all sats working
                obj.gs_func = true(size(ground_stations.name)); % initailize
all groundstations working
                obj.exp_ft = 1;
                obj.sat_type = sat_failure_type;
            end

            % Simulated Failure Model which will update the functionality
            % matrix
            function [obj] = updateFunctionality(obj,sim_clock)
```

```
                t = sim_clock.time_elapsed;
                dt = sim_clock.increment;
                ctr = t/dt;

                p_t = 1 - getReliability(t,1,obj.sat_type); % reliability for
one satellite at time t
                obj.exp_ft(ctr+1) = obj.exp_ft(ctr)*(1-p_t);

                for i = 1:length(obj.sats_func(:))
                    if (obj.sats_func(i)==1) && (rand < p_t)
                        obj.sats_func(i) = 0;
                    end
                end
            end
        end
    end
```

```
    % getReliability.m
    function [R_t] = getReliability(t,n,sat_type)
    % Given the time of operation t (seconds), and number of independent nodes
n,
    % calculates the failure rate of the path in failures/yr. Model is valid
up
    % to 3.7 years.

    % Uses the reliability model from J. Saleh in Spacecraft Reliability and
    % Multi-State Failures, pg 61, Table 4.8, Small-LEO parameters. This is a
    % 2-Weibell mixture model of small LEO satellite end of life failures

    if sat_type == "Small_LEO" % Small satellites are characterized as
<=500kg ,LEO defined as apogee and perigee up to 2000km (data from 371 satellites)
        alpha = 0.9759; % mixture fraction
        beta1 = 0.1822; % beta parameter of first Weibull (infant mortality)
        beta2 = 12.4386; % beta parameter of second Weibull (wear-out)
        theta1 = 1167882377.03; % theta parameter of first Weibull
        theta2 = 3.45; % theta parameter of second Weibull
    elseif sat_type == "Medium" % Medium satellites are characterized as 500-
2500kg (data from 371 satellites)
        alpha = 0.0128;
        beta1 = 1.0355;
        beta2 = 1.5336;
        theta1 = 0.20;
        theta2 = 103.08;
    elseif sat_type == "Large_LEO" % Large satellites are characterized as
>2500kg (data from 180 satellites)
        alpha = 0.9559;
        beta1 = 0.7840;
        beta2 = 1.4070;
        theta1 = 65.82;
        theta2 = 1.04;
    elseif sat_type == "Large_GEO" % GEO defined as near-circular orbit with
apogee and perigee at about 36000km (data from 279 satellites)
        alpha = 0.9057;
        beta1 = 0.4154;
        beta2 = 5.0600;
```

```
        theta1 = 46551.01;
        theta2 = 10.21;
    else % for test case of no failures, return value of R_t=1
        R_t = 1;
        return
    end

    u1 = log(theta1);
    b1 = 1/beta1;
    u2 = log(theta2);
    b2 = 1/beta2;
    y = log(t/36000000/24/365); % convert time from seconds to years

    R_t = (alpha*exp(-exp((y-u1)/b1)) + (1-alpha)*exp(-exp((y-u2)/b2))).^n; %
2-Weibull mixture reliability of n components

    end
```

```
% Constellation_Network.m
% This clasdef defines a full satellite constellation network, including
% both the satellites and the groundstations. It indicates operational
% communication nodes (satellites and groundstations), the network
% topology, and the weights of each edge in the network graph.

classdef Constellation_Network
    properties
        num_sats_tot
            % The total number of satellites in the entire constellation
        num_gs
            % number of groundstations in the constellation
        lat_limit
            % [rad] the highest latitude at which satellites maintain
            % inter-planar satellite connections
        gs_min_elev
            % [rad] the minimum elevation required for groundstation to
make
            % contact with satellites
        max_num_GSL
            % maximum number of groundstation-to-satellite links for a
            % single groundstation. Select the nearest satellites to link
        GS_to_GS_band
            % output data bandwidth (Gbps) to be delivered to each of the
            % other groundstations
        adj_matrix
            % This adjacency matrix shows the connectivity of all
            % the network nodes in the system. Nodes are numbered such
that
            % the satellites are first (going through the columns, from
            % left to right), and then the groundstations (in the same
            % order as they are in GroundStations).
    end
    methods
        % initialization constructor
```

```
            function obj = Constellation_Network(sats_func, gs_func,
satellite_constellation, ground_stations, lat_limit, gs_min_elev, max_GSL,
GS_to_GS_band)
                if nargin ~= 0
                    obj.num_sats_tot                                    =
satellite_constellation.n*satellite_constellation.m;
                    obj.num_gs = length(ground_stations.name);
                    obj.lat_limit = lat_limit;
                    obj.gs_min_elev = gs_min_elev;
                    obj.max_num_GSL = max_GSL;
                    obj.GS_to_GS_band = GS_to_GS_band;
                    obj     =     obj.updateAdjMatrix(sats_func,     gs_func,
satellite_constellation.sat_rv, ground_stations.gs_pos);
                end
            end

            function [obj] = updateAdjMatrix(obj, sats_func, gs_func, sat_pos,
gs_pos)
                obj.adj_matrix = zeros(obj.num_sats_tot + obj.num_gs);
                n = size(sat_pos,1); % number of satelites per orbit
                m = size(sat_pos,2); % number of orbits in the constellation

                for j = 1:m
                    for i = 1:n
                        if sats_func(i,j)
                            satID = i+(j-1)*n;
                            if i~=n
                                foreNode = satID+1;
                            else
                                foreNode = satID-n+1;
                            end
                            if sats_func(foreNode)
                                edge_nodes = sort([satID, foreNode]);
                                obj.adj_matrix(edge_nodes(1),edge_nodes(2)) =
1;
                            end

                            if j~=m
```

```
                                xSat = sat_pos{i,j}(1); ySat = sat_pos{i,j}(2);
zSat = sat_pos{i,j}(3);
                                deltaSat = atan(zSat/sqrt(xSat^2+ySat^2)); %
[rad] declination of satellite satID

                                rightNode = satID + n;
                                xRight = sat_pos{rightNode}(1); yRight =
sat_pos{rightNode}(2); zRight = sat_pos{rightNode}(3);
                                deltaRight                                  =
atan(zRight/sqrt(xRight^2+yRight^2)); % declination of Right link [deg]
                                if          sats_func(rightNode)          &&
(abs(deltaSat)<obj.lat_limit) && (abs(deltaRight)<obj.lat_limit)
                                    edge_nodes = sort([satID, rightNode]);

obj.adj_matrix(edge_nodes(1),edge_nodes(2)) = 1;
                                end
                            end
                        end
                    end
                end

                q_taken = []; % the satellites which are already linked to a
groundstation. A satellite can only link to one groundstation
                gs_ISLs = cell(obj.num_gs,1);
                for k = 1:obj.num_gs
                    r = gs_pos(k,:);
                    for q = 1:obj.num_sats_tot % do not let sats connect to
multiple groundstations
                        if any(q_taken==q) || ~sats_func(q)
                            continue
                        end
                        s = sat_pos{q}(1:3);
                        r_s = s - r;
                        cosTheta = dot(r,r_s)/(norm(r)*norm(r_s));
                        Theta = acos(cosTheta);
                        if Theta < (pi/2-obj.gs_min_elev)
                            gs_ISLs{k} = cat(1,gs_ISLs{k},[q,norm(r_s)]);
                            q_taken = cat(1,q_taken,q);
```

```
                        end
                    end
                end

                for k = 1:obj.num_gs
                    if size(gs_ISLs{k},1)<=obj.max_num_GSL
                        for p = 1:size(gs_ISLs{k},1)
                            edge_nodes                                          =
sort([k+obj.num_sats_tot,gs_ISLs{k}(p,1)]);
                            obj.adj_matrix(edge_nodes(1),edge_nodes(2)) = 1;
                        end
                    else
                        gsk_ISLs_ordered = sortrows(gs_ISLs{k},2);
                        for p = 1:obj.max_num_GSL
                            edge_nodes                                          =
sort([k+obj.num_sats_tot,gsk_ISLs_ordered(p,1)]);
                            obj.adj_matrix(edge_nodes(1),edge_nodes(2)) = 1;
                        end
                    end
                end
            end
        end
    end
```

```
% Performance_Analysis.m
% Thiss classdef defines the performance metrics used to analyze the
% constellation network.

classdef Performance_Analysis
    properties
        sat_IDs
            % array of satellite IDs
        gs_IDs
            % array of groundstation IDs
        node_thru
            % num_sats x 3 matrix with the satellite ID, total bandwidth
            % in, and total bandwidth out, for each satellite
        gs_delivered
            % num_gs x num_gs cell matrix of the delivered data rate from
            % one gs to each of the other groundstations
        tot_thru_frac
            % total throughput fraction of the network
    end
    methods
        % initialization constructor function
        function obj = Performance_Analysis(failure_model, GS_out,
routing_algorithm)
            obj.sat_IDs = routing_algorithm.sat_IDs;
            obj.gs_IDs = routing_algorithm.gs_IDs;
            obj    =    obj.getNodeThru(failure_model.exp_ft,    GS_out,
routing_algorithm.routing_paths);
        end

        % function to create a table of satellite nodes with the total
        % in/out data bandwidth, and a cell matrix of the groundstations'
        % delivered bandwidth to each of the other groundstations
        function [obj] = getNodeThru(obj, exp_ft, GS_out, comms_paths)
            node_thru_table                                              =
[[obj.sat_IDs';obj.gs_IDs'],zeros(length(obj.sat_IDs)+length(obj.gs_IDs),2)]; %
first column is the node ID number, second column is total data in, third column
is total data out
```

```
                gs_delivered_UpperMatrix                                    =
zeros(length(obj.sat_IDs)+length(obj.gs_IDs));
                for i = obj.gs_IDs(1:end-1) % origin GS ID (two-way comms)
                    for j = (i+1):obj.gs_IDs(end) % destination GS ID
                        path_ij = comms_paths{i,j};
                        if isempty(path_ij)
                            continue;
                        end
                        node_xm1_out = GS_out;
                        for k = 2:(length(path_ij)-1)
                            node_x = path_ij(k);
                            node_x_in = node_xm1_out; % no loss in the signal
transmission between two nodes
                            node_x_out = node_x_in*exp_ft(end); % loss through
a satellite node due to expected failures
                            node_thru_table(node_x,2:3)                     =
node_thru_table(node_x,2:3) + [node_x_in, node_x_out];
                            if k==(length(path_ij)-1)
                                gs_delivered_UpperMatrix(i,j) = node_x_out;
                            end
                            node_xm1_out = node_x_out;
                        end
                    end
                end
                obj.node_thru = node_thru_table;
                obj.gs_delivered      =      gs_delivered_UpperMatrix      +
gs_delivered_UpperMatrix'; % data delivered from row groundstations to column
groundstations
                obj.tot_thru_frac                                           =
sum(sum(obj.gs_delivered))/(2*GS_out*sum(1:length(obj.gs_IDs)-1));   %   total
throughput fraction of the network
            end
        end
    end
```

REFERENCES

Ash, J., and Newth, D. (2007). Optimizing complex networks for resilience against cascading failure. *Physica A: Statistical Mechanics and its Applications*, 673-683.

Ayers, M. L. (2012). *Telecommunications System Reliability Engineering, Theory, and Practice.* Hoboken: John Wiley & Sons, Inc.

del Portillo, I., Cameron, B. G., and Crawley, E. F. (2019). A technical comparison of three low earth orbit satellite constellation systems to provide global broadband. *Acta Astronautica*, 123-135.

del Portillo, I., Cameron, B., and Crawley, E. (2018). Ground Segment Architectures for Large LEO Constellations with Feeder Links in EHF-bands. *IEEE*, 1-14.

Ekici, E., Akyildiz, I. F., and Bender, M. D. (2001). A Distributed Routing Algorithm for Datagram Traffic in LEO Satellite Networks. *IEEE/ACM Transactions on Networking*, 137-147.

Ereau, J.-F., and Saleman, M. (1996). Modeling & Simulation of a Satellite Constellation based on Petri Nets. *Annual Reliability and Maintainability Symposium* (pp. 66-72). Las Vegas: IEEE.

Farserotu, J., and Prasad, R. (2000). A Survey of Future Broadband Multimedia Satellite Systems, Issues and Trends. *IEEE Communications Magazine*, 128-133.

Foust, J. (2019, July 1). *Starlink failures highlight space sustainability concerns*. Retrieved from Space News: https://spacenews.com/starlink-failures-highlight-space-sustainability-concerns/

Harris, M. (2019, October 16). *Amazon Reports Collision Risk for Mega-Constellation of Kuiper Internet Satellites*. Retrieved from IEEE Spectrum: https://spectrum.ieee.org/tech-talk/aerospace/satellites/amazon-reports-collision-risk-for-its-megaconstellation-of-kuiper-internet-satellites

Howell, K. (2018). Orbital Perturbations. *AAE 590 Orbital Analysis Lecture*. West Lafayette, IN, USA: Purdue University, School of Aeronautics and Astronautics.

Humboldt State University. (2014). *Atmospheric Absorption & Transmission*. Retrieved from Humboldt State Geospatial Online Introduction to Remote Sensing: http://gsp.humboldt.edu/OLM/Courses/GSP_216_Online/lesson2-1/atmosphere.html

Ingram, Ingram, M. A., Barott, W. C., Popovic, Z., Rondineau, S., Langley, J., and Romanofsky. (2005). *LEO Download Capacity Analysis for a Network of Adaptive Array Ground Stations* . NASA.

Jehn, R., Agapov, V., and Hernandez, C. (2005). The situation in the geostationary ring. *Advances in Space Research*, 1318-1327.

Li, S.-Y., and Liu, C. H. (2002). An Analytical Model to Predict the Probability Density Function of Elevation Angles for LEO Satellite Systems. *IEEE Communications Letters*, 138-140.

Mortari, D., Wilkins, M. P., and Bruccoleri, C. (2004). The Flower Constellations. *Journal of the Astronuatical Sciences.* Department of Aerospace Engineering, Texas A&M University.

OneWeb. (2013). *OneWeb Non-Geostationary Satellite System - Technical Information to Supplement Schedule S.* Washington, D.C.: Federal Communications Commission.

Saleh, J. H., and Castet, J.-F. (2011). *Spacecraft Reliability and Multi-State Failures.* West Sussex: John Wiley & Sons Ltd.

Saleh, J. H., Hassan, R., Torres-Padilla, J., Hastings, D., and Newman, D. (2005). Impact of Subsystem Reliability on Satellite Revenue Generation and Present Value. *Journal of Spacecraft and Rockets*, 1122-1129.

Schneiderman, B. (2019, March 4). *The Next Wave: Low Earth Orbit Constellations*. Retrieved from Satellite Markets & Research: http://satellitemarkets.com/news-analysis/next-wave-low-earth-orbit-constellations

Spencer, D. (2019, March). Telecommunications Link Budget. *AAE 590 Space Flight Operations Lecture*. West Lafayette, IN, USA: Purdue University School of Aeronautics and Astronautics.

Wang, J., Li, L., and Zhou, M. (2006). Topological dynamics characterization for LEO satellite networks. *Computer Networks*, 43-53.

Wood, L. (2001). Internetworking with Satellite Constellations. *Thesis submitted to the University of Surrey for the degree of Doctor of Philosophy*. Guildford, United Kingdom: University of Surrey.

Wood, L. (2003). Satellite Constellation Networks. *Internetworking and Computing Over Satellite Networks*, 13-34.

Wood, L., Pavlou, G., and Evans, B. (2001). Effects on TCP of Routing Strategies in Satellite Constellations. *IEEE Communications Magazine*, 172-181.