

Modelling Low Earth Orbit Constellations for Networking

Joseph McGuchan

May 15, 2019

I, Joseph Law McGuchan of King's College, being a candidate for Part II of the Computer Science Tripos, hereby declare that this dissertation and the work described in it are my own work, unaided except as may be specified below, and that the dissertation does not contain material that has already been used to any substantial extent for a comparable purpose.

Signed

Date

Modelling Low Earth Orbit Constellations for Networking

Contents

1	Introduction	5
1.1	What is Starlink?	5
1.2	Why Focus on Starlink?	6
1.3	Why Create a Visualisation?	7
1.4	Mark Handley's Work	7
1.5	What I Will Be Using	7
2	Preparation	7
2.1	Starting Point	7
2.2	The Structure of Starlink	8
2.3	Modelling Sattelites in Orbit	9
2.4	Unkown Variables	10
3	Implementation	11
3.1	Repository Overview	11
3.2	Coding for Visualisation and Efficent Algorithms	11
3.3	Modelling the Network	12
3.3.1	Difficulties of Modeling Orbits	12
3.3.2	Representing Links	12
3.3.3	Patterns used	13
3.3.4	Dijkstra's Algorithm	14
3.3.5	Calculating the Latency Across a Path	14
3.3.6	Representing a Constellation	14
3.3.7	Efficiency vs Usalibitly	15
3.4	Visualization	15
3.4.1	How to Use the Exe	16
3.4.2	The Motion of the Player	16
3.5	Initial Observations of the Network	17
3.5.1	The Structure of the Network	17
3.5.2	Movement of Satellites Relative to Each Other	17
3.5.3	Periodicity in Latencies	18
4	Evaluation	21
4.1	Developing Linking Methods for the Network	23
4.2	Comparing Linking Methods	23
4.2.1	Comparing Linking Methods On Latency	23

4.2.2	Comparing Linking Methods On Fault Tolerance	23
4.2.3	Comparing Linking Methods On High Demand	24
4.3	Visualising the Linking Methods at Work	24
4.4	Direction and Focus	24
4.5	Extension	25
5	Conclusion	25
A	Appendix A	26

1 Introduction

SpaceX are planning to launch a constellation of 4,425 low Earth orbit communication satellites in the next few years. The objective of this constellation, called Starlink, is to provide low-latency internet connection across the world.

The satellites in this network will be in constant motion, not just relative to the ground, but relative to one another, creating a network with a constantly changing topology and associated latencies. The question of how to structure such a network, and what the resultant latencies of said network will be, has not been thoroughly explored, but it will become increasingly relevant as more and more companies build similar constellations. If these Constellations prove to provide significant gains in latency while providing competitive bandwidth, they might render previous submarine optical cables obsolete.

My goals are:

1. To create visualizations of the SpaceX constellation.
2. To experiment with different topologies of the SpaceX network and test their associated latencies.

1.1 What is Starlink?

The purpose of Starlink is to provide low latency internet connections, as of 21/11/18, there are two companies offering satellite internet services, Excede[?], and Hughes, whose 9202 BGAN Land Portable Satellite Terminal offers connection speeds up to 464kbps[2]. These companies largely target

domestic use in rural areas which don't have a faster coverage, and corporations, providing internet connections to airplanes and cargo ships. Currently, Satellite Internet connection is a last resort, something turned to when conventional means of connection are not available, Starlink intends to invert this, turning satellite internet into the premium option.

The difference between Starlink and currently existing brands is that current brands use geostationary satellites [9], while Starlink will use LEO satellites. The significantly shorter distance will create much shorter paths for signals. On top of this, SpaceX will be utilizing laser communication between satellites, as opposed to the competitors who have little to no intra-satellite communication. In the vacuum of space, lights travels 47% faster than in glass [8]. Therefore, in theory, a LEO network utilising lasers would achieve latencies far lower than that by even the best terrestrial fiber optic connections over long distances.

1.2 Why Focus on Starlink?

Starlink is only one of a number of different LEO internet networks that have been proposed, so why should I focus on it's topology? Ideally, I would like the conclusions of my study to be generalized to many other LEO constellations. However, constellations are approved by the FCC on a case-by-case basis, using a complicated and changing system of legislation, it's not easy to know what a normal network looks like. By analyzing the properties of a network of my own design, I run a much greater risk of coming to conclusions that cannot be generalized, as I am not working on an FCC-approved constellation.

What about the constellations of other companies?

By analyzing the properties of Starlink, we are analyzing a network that is confirmed legally and scientifically plausible.

But there is another reason to investigate Starlink. As the largest of most high-profile attempts to build a LEO internet backbone, Starlink represents the most significant competitor to other emerging satellite communications networks and the one more likely to become dominant in future years. By theorising about it's properties now, we can prepare ourselves for the changes Starlink might pose to communications in the upcoming years.

1.3 Why Create a Visualisation?

When it comes to understanding a network such as starlink, a visual description is incredibly valuable. By visualizing the network we can develop an intuition for how it operates, and use that intuition to develop ideas for new algorithms and structures for testing.

Creating a visualization also poses a minimal additional cost on my part, as to accurately model the latencies between base stations I will need to simulate satellite positions and links anyway.

1.4 Mark Handley's Work

My initial proposal for the project was inspired by the findings of an existing study done by Mark Handley, since proposing the project, Mark Handley has made his results publicly available[6], so my new goal will be to replicate and expand upon his findings.

1.5 What I Will Be Using

To create the visualization I will be using the open-source game engine Godot. Using a game engine struck me as the simplest way to create a visualization tool, and Godot, being powerful, open-source, and capable of running easily on many devices, seemed like the ideal choice.

For the parts of code related to visualization I will be using godot's build in script GDScript, which is easy to use and specifically designed to integrate well with the godot engine. For parts of the code related to simulation I will use C#, which godot is compatible with, and which offers significance performance improvements over gdscript.

2 Preparation

2.1 Starting Point

Most of the information I will be using for the structure of the SpaceX network comes from their application to the FCC on March 29th, 2018[4], and their technical attachment[5], which outlines the various orbital spheres, planes within each sphere, and various other details of the constellation. I

Figure 1: The layout of Starlink

SPACEX SYSTEM CONSTELLATION					
Parameter	Initial Deployment (1,600 satellites)	Final Deployment (2,825 satellites)			
Orbital Planes	32	32	8	5	6
Sattelites per Plane	50	50	50	75	75
Altitude	1150km	1110km	1130km	1275km	1325km
Inclination	53°	53.8°	74°	80°	70°

will also be building on insights gleaned from Handley’s original research. The insights that I will be building off of are:

SpaceX have already sent up two test sattelites, and according the Elon Musk, they are working very well, providing a latency of only 25ms[3].

2.2 The Structure of Starlink

There is a lot we do not know about Starlink, but this is what we can infer from SpaceX’s application to the FCC[4],

Starlink is a constellation of 4,425 LEO sattelites with a further proposed.

We will be examining only the 4,425 sattelites proposed for the first implemantation of Starlink, as these intend to provide a viable network by temselves, and additional sattelites might overwhelm my visualisation techniques.

In it’s application to the FCC SpaceX was required to state any notable debris that might not burn up if a satellite were to descend from orbit. Among the components listed where 5 silicon carbide “communication components”, silicon carbide is used in mirrors for laser communication links, and we can therefore conclude that SpaceX’s sattelites will have, at most, 5 available links to form with nearby sattelites.

We also know information about how Starlink sattelites will communicate with ground stations, a base station may only connect with a satellite at an inclination less than 40°from the vertical using Radio Frequency (not laser) connections. Connections will be stronger to sattelites directly overhead, however as Starlink does not provide much information in exactly how their intra-satellite communications work, speculations on bandwidth will

not prove to be helpful.

2.3 Modelling Sattelites in Orbit

Typically, an orbit around the surface of the earth is described by:

Distance Above the Surface

Inclination This is the angle between the orbital plane, and the equatorial plane, (the plane on which the equator lies).

Longditudonal Offset If inclination is greater than 0° , then the orbital plane and equatorial plane will intersect at a line, the angle between this line and the plane described by the great circle at longditute 0° is the longditudonal offset. In other words, the Longditudonal offset is the Londitudute of the point where this line passes through the surface of the earth.

Eccentricity The eccentricity coefficient describes the “squashness” of the orbit. For the time being, we will be ignoring this...

Retrograde A retrograde orbit is one that goes against the rotation of the earth, typically these are described by orbits with a longditudonal offset greater than 180° . It is significantly more expensive to put a sattelite into a retrograde orbit, and is essentially never done.

Furthermore, an individual satellite’s position in an orbit can be described by:

True Anomaly The angle between the line drawn from the origin to the satellite, and the line drawn from the origin to the point on the satellite’s orbit of longditude 0.

Phase Offset This is the satellite’s true anomaly at time 0.

In our model, true anomaly is the only variable that changes. The way it changes is uniquely determined by the other 6 variables. The rate of change of true anomaly, the angular velocity, is given by

$$\sqrt{\frac{GM}{r^3}}$$

Where r is the distance from the origin, or altitude + radius of the Earth.

Note that while altitude is constant (eccentricity = 0) velocity is unchanging, and true anomaly can be described as a linear function of time.

The position of a satellite is calculated by taking it's true anomaly and adding it's angular velocity multiplied by the timestep (real timesince last frame update * some factor). The location is then calculated through a series of transformations performed on the true anomaly:

1. We take a vector $(r, 0, 0)$ where r is the distance above the surface + the radius of Earth.
2. We rotate this point around $(0, 1, 0)$ (the line through the poles) by the true anomaly.
3. We rotate this point around $(1, 0, 0)$ (the line from 0 longitude to 180 longitude through the equator) by the inclination.
4. We rotate this point around $(0, 1, 0)$ again by the longitudinal offset.

Because the only variable that is changed is the true anomaly, and the x , y , and z coordinates are determined by only this variable and a series of fixed variables, we do not run the same risks normal discrete physics models face when describing continuous behavior, such as the energy of the system changing, or orbital precession (the point of furthest distance moving steadily), as we will not be approximating a continuous process from a series of discrete steps but instead will be taking samples from a continuous function.

For the rest of this document, a satellite will be denoted by $x_{i,j,k}$. Describing the k th satellite on the j th orbital place in the i th orbital sphere. L gives the number of orbital spheres, with M_i giving the number of orbits in orbital sphere i , and $N_{i,j}$ giving the number of satellites in the j th orbit of the i th sphere.

2.4 Unkown Variables

While we can learn a lot from SpaceX's applications, there are still a number of variables that are left undetermined.

Phase Offset As we do not know for certain how satellites will be positioned in orbits, we also do not know how they will be positioned relative to

other orbits. The phase offset of an orbit $o_{i,j}$, $po_{i,j}$, can be described as a number from 0 to 1, where, $\forall k$, 0 indicates that $x_{i,j,k}$ will cross the equator at the same time as $x_{i,j+1,k}$, and 1 indicates that satellite $x_{i,j,k}$ will cross the equator at the same time as $x_{i,j+1,k+1}$. This number will have to be some fraction of the total number of orbits, to guarantee that the first and last orbits are properly aligned.

Sphere Offset As well as satellites being offset from one another, whole spheres can be offset too. I will only be looking at a single sphere in my study, however, so this variable will not be examined.

Link usage Each satellite will have a maximum of 4 links, however, we do not as of yet know which arrangements of links are the most optimal.

In his study Mark Handley uses two links to connect $x_{i,j,k}$ to $x_{i,j,k+1}$ and $x_{i,j,k-1}$. He then uses the other two to connect $x_{i,j,k}$ to $x_{i,j+1,k+h_i}$ and $x_{i,j-1,k-h_i}$, where h_i is a variable that describes how "diagonal" these sideways links should be. For $j = 0$ and $j = number_of_orbits$, $x_{i,0,k}$ connects to x_i ,

In general, a fixed-link linking method can be described by an $n/2$ by 2 matrix, where n is the number of links each satellite has free. In this case, that means a 2x2 matrix.

3 Implementation

3.1 Repository Overview

3.2 Coding for Visualisation and Efficient Algorithms

This implementation chapter is divided into two main sections, the first section will go over how I represented the constellations and the code I wrote to perform tests on this model, the second section will go over the executable file, its structure, and how to use it. While these sections are separate in my report, they were developed concurrently, and are connected to each other. As I visualised the constellation, I was able to gain a better understanding of its structure, this directed the properties of the network I wanted to examine, which in turn effected the tests I wanted to create and so the algorithms I needed to implement. Furthermore as Godot is designed to model 3d environments it provides a number of functions that in turn influenced how I

wrote my algorithms, most rotably Godot provides a simple framework for multithreading through a `_process` funtion that any calss extending `Node` can override and which is called on every step by the engine. Godot handles multithreading of the process function internally. This gave me the opportunity to exploit threading to improve the efficiency of the program without having to do any extra work. All that said, I think to is clearer to divide these two halves of the program when talking about my implementation.

After these two sections will be one more outlining observations that were made of the network's behaviour during my experimenting on it, and how these observations lead to further development in my linking algorithms.

3.3 Modelling the Network

3.3.1 Difficulties of Modeling Orbits

On Implementation, I found that my planned method of implementing orbits was flawed, being able to simulate only 500 satellites in motion before slowing down. Because of this, I changed to a precomputed method. In this method, orbits are precomputed as a number of transformations, and the position of satellites is calculated by interpolating adjacent transformations, this sacrifices some accuracy, but vastly increases the computation speed. It is important that orbits be described by transformations (a combination of a point and rotation) so that satellites can be oriented to ways have the earth below them from their reference frame. This makes a umber of computations later on easier.

3.3.2 Representing Links

There are 3 primary ways to represent a graph for algorithms:

Adjacency Matrix An $N \times N$ matrix of values representing all possible connections. Obviously inefficient for a sparse set of connections like this.

Adjacency List Associate each Satellite with a list of the vertices it is connected to. Space efficient, but creates difficulty in running Dijkstra's algorithm.

Edge List Create a list of edges that refer to vertices. This is useful for algorithms that require iterating through all edges like Dijkstra's algo-

rithm. But to find the adjacent satellites to any given satellite requires searching through the whole list.

Initially, I used adjacency lists, this lead to a number of problems.

- Some algorithms require a list of all edges, which is very hard to obtain with adjacency lists.
- When drawing edges in the visualization, each would be drawn twice, for each vertex they were associated with.
- When updating links, values such as the distance between two verticies had to be computed twice for each vertex.

Instead I created a new class of objects representing links, and gave each vertex a reference to the links associated with it.

In Godot, all drawn objects require one "parent" (different to class inheritance) which is already a drawn object. Children inherit the position and rotation of their parents, and rotating a group of objects is significantly more efficient than rotating a single object alone. Because of this, backwards and forward links, which do not change distance or position relative to the satellite, should be made children of the satellite, and they will then never have to be updated. Because of this, all links have a primary and secondary satellite. Which the link always being the child of the primary satellite.

3.3.3 Patterns used

Due to the extensive combustibility of the network, I elected to use the Template Method pattern in three separate locations.

- To describe different linking methods.
- To describe various different ways of colorizing the network to highlight certain properties.
- To describe various tests to be applied to the network.

I also utilized a variant of the Builder pattern for describing constellations and tests. The ConstellationDescription class is initialized with a description of a constellation similar to what was given above, including the tests and colouring method I want to use on the class. This allowed me to easily categorize and load new constellations.

3.3.4 Dijkstra's Algorithm

I used Dijkstra's algorithm in order to calculate shortest paths around the network, with the modification to allow for quickly retrieving the shortest path when used. Somewhat surprisingly, C sharp doesn't come with a priority queue, so I used one from <https://visualstudiomagazine.com/Articles/2012/11/01/Priority-Queues-with-C.aspx?Page=2>. And used comparisons from <https://github.com/BlueRaja/High-Speed-Priority-Queue-for-C-Sharp>.

3.3.5 Calculating the Latency Across a Path

There are 4 factors that effect the latency of a path, some of which are easy to model, and others are significantly more tricky.

Processing delay The time it takes router to process the packet header, this is typically insignificant and will be assumed to be 0.

Queuing delay The time the packet spends in routing queues, this is dependant on the load on the network and is therefore not easy to predict. I will be calculating lanency assuming the best case scenario, which is no queing delay.

Transmission delay The time it takes to push the packet onto the link. This is of course dependant on the size of the packet, but for the smallest packet sizes this value is negligible.

Propagation delay The time the signal spends in transmission, since all connections are of the form of electromagnetic waves this is simply the path length divided by the speed of light.

Because of this, the only calculation needed to calculate needed to get the latency of a path is to divide it's length by the speed of light in a vaccum. To calculate the round trip time (RTT), which is typically used when advertising linking systems, we simply multiply this value by two.

3.3.6 Representing a Constellation

Constellations in my database are described in the a simple format that allows for much portability, this figure contains how to describe one theory or how the spaceX constellation might be arranged.

Figure 2: An example for how a constellation might be described.

Number of Orbital Spheres	5				
Sphere Number	1	2	3	4	5
Altitude	1150km	1110km	1130km	1275km	1325km
Inclination	53°	53.8°	74°	80°	7°
Number of Orbits	32	32	8	5	6
Satellites per Orbit	50	50	50	75	75
Offset Between Orbits					
Offset of Sphere					
Linking Method	X	X	X	X	X

$$X = \text{Handley}(-1)$$

Notice the addition of offsets and linking methods, these are necessary to fully describe a constellation, but are not given in an FCC application.

3.3.7 Efficiency vs Usability

To do this requires the creation of numerous abstract classes and wrappers that would simplify more complex objects. There are two kinds of vertex, base stations and satellites, and two kinds of link, those between satellites and those between a satellite and base station. Each of these are children of general Vertex and Link classes, and it is these classes that are used by the various tests and algorithms that run on the network. Initially my objective was to prioritise efficiency at all costs, but as I continued to develop the code, it became necessary to implement redundancies and inefficient storing methods, like lists, in order to make the code more easily usable. I also created a generic Graph object that contained only a list of Vertices and a list of Links, and a VertexPath object that allowed for easy manipulation and testing of paths.

3.4 Visualization

For the earth texture I used a creative commons equirectangular map of the Earth. Equirectangular means that x and y values correspond to longitude

Figure 3: Keybindings.

Move Forward	W, Up Arrow
Move Backward	S, Down Arrow
Move Up	Q, Comma Key
Move Down	E, Full Stop Key
Move Left	A, Left Arrow
Move Right	D, Right Arrow
Restart Simualtion	R
Toggle Free Mouse	Esc
Next Test	X
Previous Test	Z

and latitude. It was important that the map be equirectangular as this is the format that is compatible with being mapped to a sphere. [13]

3.4.1 How to Use the Exe

If you would like to see the program that was used to generate these images and results, simply download and run the exe hosted here

The exe will create files for test results in the same location as it is ran, as well as printing to the command line the results of tests as they are written.

It is reccomended that you run in Windows 10, as it is not garunteed to work in other operating systems.

You can navigate the simulation using the mouse, along with arrow keys or WASD. Press R to reset the whole simualtion. Use X and Z to move between different tests. Press ESC to toggle the mouse between captured (for moving the camera) and free.

All the tests run at 60x realtime, which I found to be a very good time-factor for observing overarching trends in the network.

There are a total of

3.4.2 The Motion of the Player

One of the few pieces of code written in Godot's native language GDScript is the code that controls user interaction with software. It made sense to write this in GDScript as it is a much simpler language than C#, and because

a very simple tutorial of first person motion was available in GDScript[12]. However this code required some adaptation to make it work in 3D, primarily the removal of a number of unnecessary features, and the addition of the ability to move up and down.

The code works using a "head" which is attached to a "neck". Left-right motions of the mouse are translated to left-right rotations of the neck, while up-down motions of the mouse are translated into up-down rotations of the head. keeping the head and neck separate mean that the player cannot turn themselves upside down. Motion is done relative to the head's view, with gentle smoothing applied to motion to give a more natural feeling.

3.5 Initial Observations of the Network

While working on the program I made a few notable observations of the network. In this section I go through each of these observations and outline how it fed into what I was doing.

3.5.1 The Structure of the Network

One of the most notable observations of the network was it's unique, layered grid structure. To highlight this I applied a different hue to each orbit 3.5.1. As you can see, for a given point near the equator, there are in fact two very different kinds of Satellite moving overhead, those going North-East, and those going South-East. Towards the poles, on the other hand, all satellites are moving Eastward. This means that Satellites that pass very close to one another on the equator can end up on opposite sides of the Earth close to the poles. As described below, it is very likely that Satellites will only link to other satellites moving the same direction as them, and in my model I have reflected that. This means that for a given point on earth there will be two almost seperate grids moving overhead, one grid of satellites moving North-East, and another of satellites moving South-East.

3.5.2 Movement of Satellites Relative to Each Other

I decided to track the path of a satellites neighbors relative to it and it's direction of movement over the course of an orbit around the Earth. In 3.5.2 the paths traced by nearby satellites is given. The satellites directly ahead of and behind a given satellite stay in a steady position relative to it and it's

Figure 4: The Starlink Constellation with a different hue appllied to each orbit.

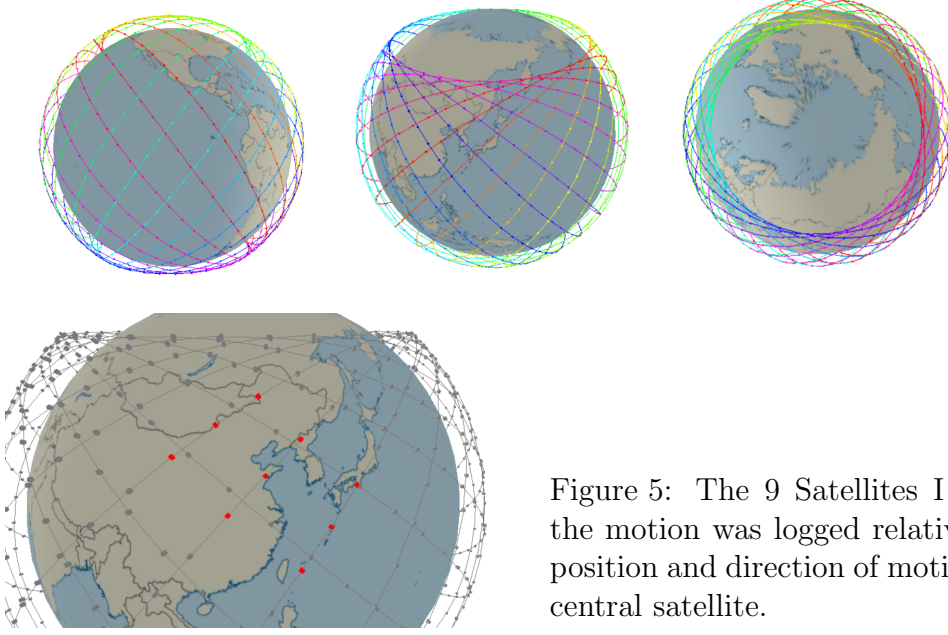


Figure 5: The 9 Satellites I tracked, the motion was logged relative to the position and direction of motion of the central satellite.

direction of motion, while those in adjacent orbits slowly figure eights. You can see in 5 that there are other nearby satellites in different orbits, however these satellites are moving South while the highlighted satellites are moving North, meaning that they will be moving by far too quickly for the main satellite to track them as they pass. [6]

These results show that satellites in adjacent orbits will be moving in relatively slow and predicatable ways meaning it should be possible for a satellite to track a laser link to satellites in adgacent orbits. It also shows that links to satellites in the same orbit will be significantly easier as satellites in the same orbit maintain a constant position relative to one another.

3.5.3 Periodicity in Latencies

When initially examining latencies in the network I noticed a slow transition in the result that appeared to suggest some kind of periodicity. 3.5.3 To examine this further, I did a further test, this time running the model for a much longer time. 3.5.3 What I found was a periodicity in the latency

Figure 6: An example of the paths that neighbors take relative to a satellite and its direction of motion.

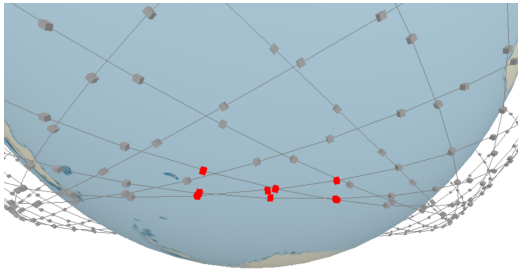
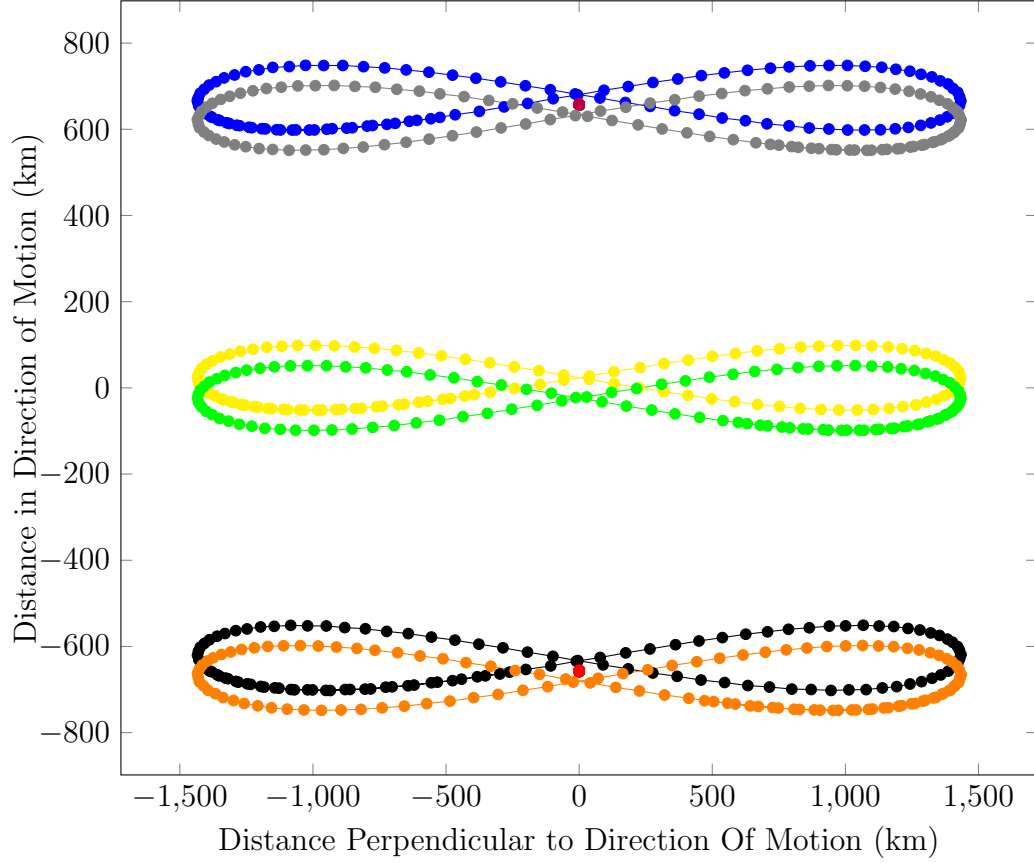


Figure 7: Towards the South Pole, the square of satellites fold over and the satellites that were once on the right of the central satellite move to the left.

Figure 8: The distance over time from a satellite to it's neighbors perpendicular to it's direction of motion and the direction towards the core of the Earth, or how far to the left or right of the satellite it's neighbors are over time.

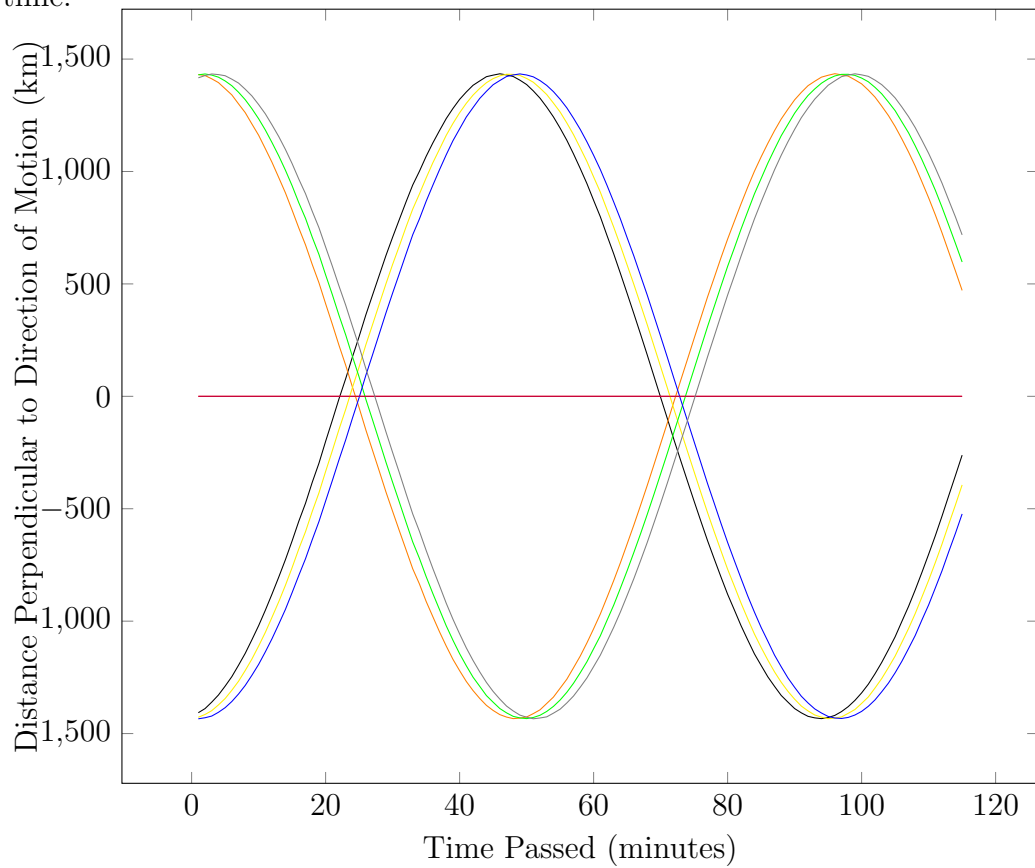
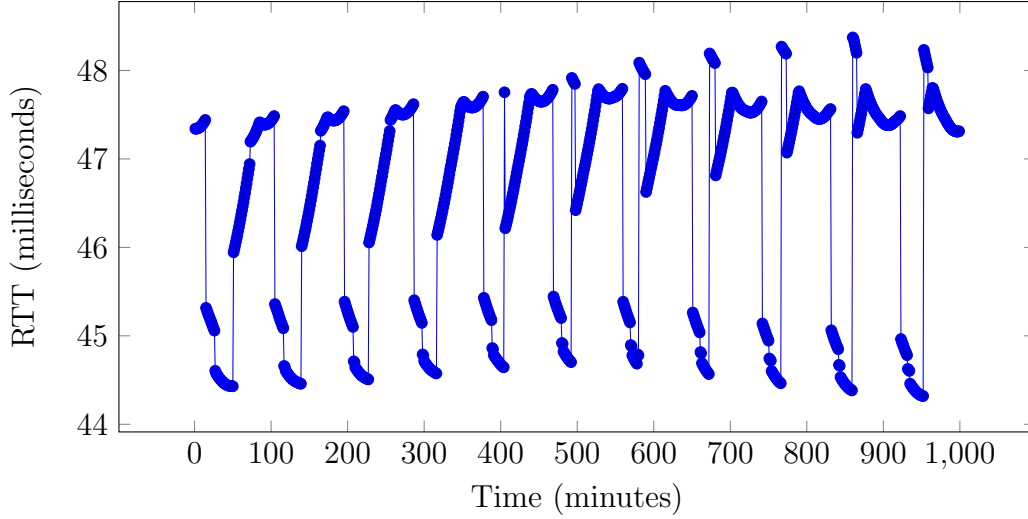


Figure 9: The Latency of the London-New York connection over time



data of almost exactly one hour. I connected this to the rotation of the earth under the starlink network, which has 24 orbits. To confirm this I ran a third test with the Earth frozen, and found the gradual changes observed initially has disappeared. 3.5.3 As far as I can tell, this hour-long variation in the latency of a path is not something that Mark Handley accounted for in his research, however to account for it in mine, all latency results for a link will be determined by taking the average of regular samples across an hour of simulated time.

4 Evaluation

My goals going into this project were:

- To develop multiple routing algorithms for this network.
- To compare these algorithms on latency, fault tolerance, and response to high demand.
- To create visualisations of these routing algorithms at work.

With success being determined if I can test three different algorithms, producing meaningful data about their latency, fault tolerance, and response under high demand along with informative visualisations of their operations.

Figure 10: The Latency of the London-New York connection over time (extended)

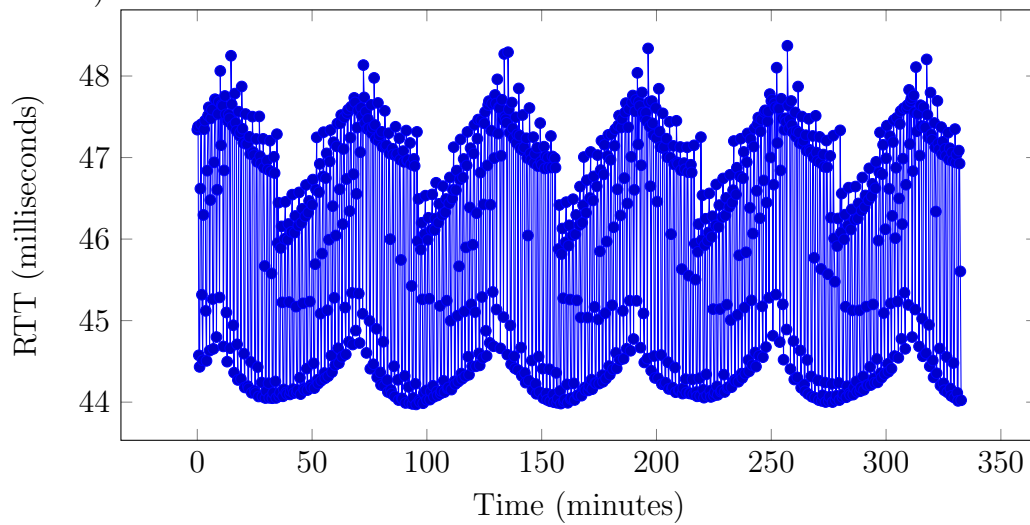
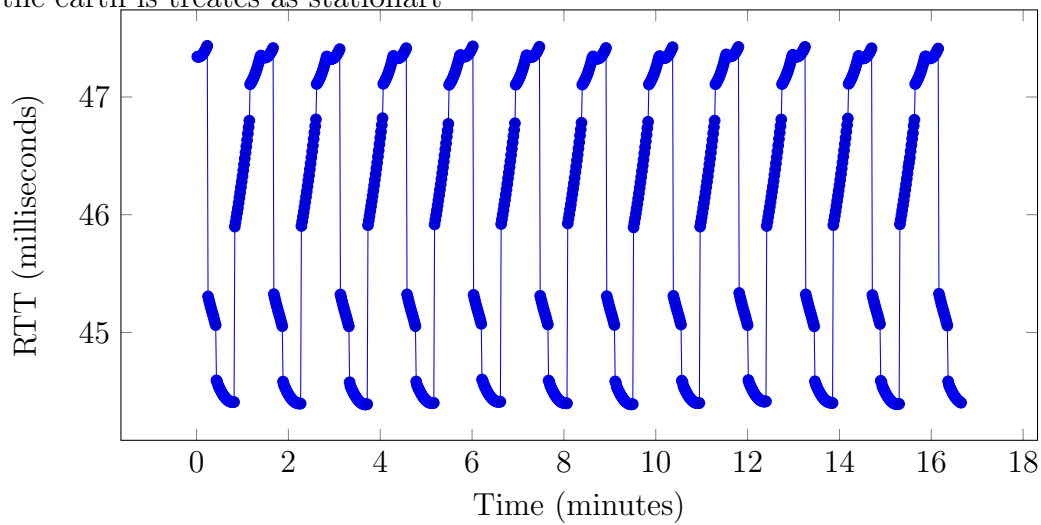


Figure 11: The Latency of the London-New York connection over time when the earth is treated as stationary



It became apparent early on that "routing algorithms" was a very ambiguous term. It is not clear how one would develop an entirely new routing algorithm to operate on this network, or whether SpaceX would choose to. Furthermore, the behaviour of the network is extremely predictable, which means that a base station can easily calculate the optimal path across the network and send a packet up with the exact satellites it needs to be passed between, this makes static routing the most likely option. What is very much undetermined is the topology of the network being routed across, specifically the linking method and the phase offset. Because of that, I chose to examine the properties of 3 linking methods instead.

4.1 Developing Linking Methods for the Network

The three linking methods I chose to examine are

•

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \textit{Handley}(0)$$

•

$$\begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} = \textit{Handley}(-1)$$

•

$$\begin{bmatrix} 2 & 1 \\ -1 & -1 \end{bmatrix} = \textit{DistantLinking}$$

The first two were variations of the linking methods used by Mark Handley, the third was chosen based off of observations of the network. Specifically I wanted to test whether having satellites "skip" nearby connections for more distant links would lead to an improvement in performance by reducing the number of hops in shortest paths.

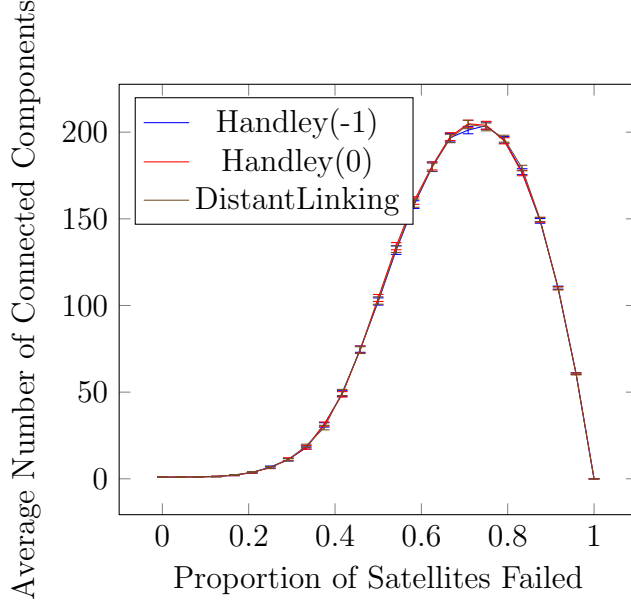
4.2 Comparing Linking Methods

4.2.1 Comparing Linking Methods On Latency

4.2.2 Comparing Linking Methods On Fault Tolerance

The first test I did for fault tolerance was a simple test of total connectivity. I deleted a certain percentage of satellites from each linking method, and

Figure 12: The number of connected components after various amounts of deletion of random satellites, using the three linking methods outlined.



then tested the total number of connected components. I did this 100 times, so that my 95% confidence interval never exceeded 4 components. What I found is that all 3 linking methods had exactly the same curve of connected components, with the expected number of connected components rising after a 20% failure rate, peaking at an 80% failure rate, and then dropping off as the number of satellites being removed from the constellation outpaced the number of new constellations being created.

4.2.3 Comparing Linking Methods On High Demand

4.3 Visualising the Linking Methods at Work

4.4 Direction and Focus

Maintaining a consistent goal has been challenging during such an open-ended project. If I were to do this project again I would more carefully choose my objectives

A number of projects in the early phases of the project had to be abandoned

done not just because of inefficiency but because new information came about that rendered them useless.

The announcement from SpaceX on that they were significantly changing the constellation had massive impacts to the project. Firstly, the new satellites would have only four links instead of five, this implied that the satellites would not have a "free" link but would instead be connected only to those ahead of and behind them. Because of this, I decided to drop the free links from my model, but doing so meant that a large chunk of code that had been written was discarded.

The knowledge that SpaceX were willing to change their designs so abruptly also had a significant impact on modelling their constellation. It is likely that the second and third phases of Starlink will be similarly changed to be in similar altitudes to phase one, however without a full understanding on what these new spheres would be, or how they would connect to phase one, it was elected to instead focus solely on the first sphere's constellation.

4.5 Extension

In Handley's original paper, he anticipated ...

5 Conclusion

I believe that I have made a number of very significant extensions to Handley's work. The discovery of the 60-minute variations in latencies is in my opinion very significant, this is something that I have not seen observed in any of the research thus far but has a significant impact on how one calculates latencies across the network.

References

- [1] <https://www.exede.com>
- [2] <https://www.hughes.com>
- [3] <https://twitter.com/elonmusk/status/1000453321121923072>

- [4] licensing.fcc.gov/cgi-bin/ws.exe/prod/ib/forms/reports/related_filing.hts?f_key=-289550&f_number=SATLOA2016111500118
- [5] https://licensing.fcc.gov/myibfs/download.do?attachment_key=1158350
- [6] <http://nrg.cs.ucl.ac.uk/mjh/starlink/>
- [7] <http://stuffin.space/>
- [8] <http://ece466.groups.et.byu.net/notes/smf28.pdf>
- [9] <https://www.hughes.com/who-we-are/resources/press-releases/hughes-high-throughput-satellite-successfully-launched-setting?locale=en>
- [10] https://space.skyrocket.de/doc_sdat/jupiter-2.htm
- [11] <file:///home/joe/Downloads/R12-ITURKA.BAND-C-0008!!PDF-E.pdf>
- [12] <https://github.com/turtletooth/GodotFirstPersonController>
- [13] https://commons.wikimedia.org/wiki/File:USGS_majplatecolor.png

A Appendix A