Report 1045

The aim of Project Loon is to ensure everyone on the planet has access to the internet, by creating a balloon-powered network. How many balloons would be required to provide balloon-powered internet coverage to all of New Zealand?

Introduction

In order to determine the optimum number of balloons required to provide full coverage of internet to New Zealand, we need to determine the area required to provide coverage to, movement of balloons in the stratosphere, the optimal placement and the maximum range of coverage of each balloon while taking into account the redundancy of the balloon network.

Firstly a model is established which demonstrates the optimal placement of balloons between each other and the ground stations. Then a simulation is used to generate the area of coverage as well as the minimum number of balloons required to cover New Zealand based on the pre-determined optimal placement of the balloons. There are possible sources of interference with the ISM band of signal which are taken into account by the simulation which are:

- Noise generated by population density. This is due to existing wireless networks and other appliances working in the 2.4-5.8GHz band.
- Population density also increases the bandwidth requirement, hence this also increases the required number of balloons in a specific area.
- Changes in terrain causing the destruction of the wireless signal due to absorption from vegetation and blocking of signal from terrain this is due to wireless signals not easily diffracting around obstacles.

Assumptions

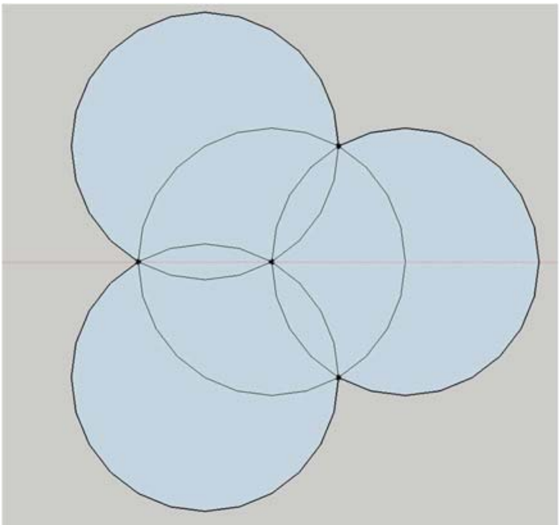
- Noise, Signals do not travel as well within densely populated areas (e.g. Cities) due to existing networks.
- When one balloon goes down due to malfunction/servicing, other balloons need to cover enough area to not disrupt service.
- Building do not have a large effect. The noise generated by devices already present in the 2.4-5.8GHz bands already provide enough interference to justify the increase in the number of balloons
- Since New Zealand is the target of this experiment, it is impossible to not have sunlight to charge the balloons
- -Aviation laws are not taken into account as quote Google:
- "Our goal is to make sure we don't surprise anyone. We're going to start at about the 40th latitude south. We'll be talking with the relevant authorities in countries where we may be flying over or looking to set up pilot programs -- e.g. Chile, Argentina, Uruguay, Australia. There are already thousands of weather balloons launched each year, so there are established processes that we follow with local aviation authorities."
- http://www.wamda.com/2013/06/project-loon-google-answers-our-questions-about-its-%E2%80%98internet-for-everyone
- -Earth is a perfect sphere except for New Zealand. This is to avoid problems in our model. This is also because the zonal region is small in comparison with the rest of the earth, so it will not affect the real result that much.
- The atmosphere of the earth rotates along with the rest of the earth, therefore the velocity of the balloons will be roughly constant in relation to the velocity of the eath.

Determining the area of coverage

The balloons will be travelling over New Zealand by following the latitudes of the Earth which means longitudes are not needed. In order to find the zonal region where the balloons will be travelling mainland New Zealand is taken to begin from the North Cape with a latitude of 34°23'S or 34.3833°S and end at the South West Cape with a latitude of 47°17'S or 47.2833°S. [1]

As the balloons will travel across Chatham Islands as it is in the same zonal region as rest of NZ, the population of 600 will receive Internet. Islands of Bounty[2], Antipodes[3], Auckland[4], The Snares[5], Three Kings[6] and Campbell[7] which are part of the extended territories of New Zealand as shown on the left are not included in the zonal region which we used for the modelling of the coverage of New Zealand, but do not have an effect as none of these are inhabited and therefore do not require internet. However the Kermadec Islands are inhabited[8], but the population of only 10 means it its significance is negligible.

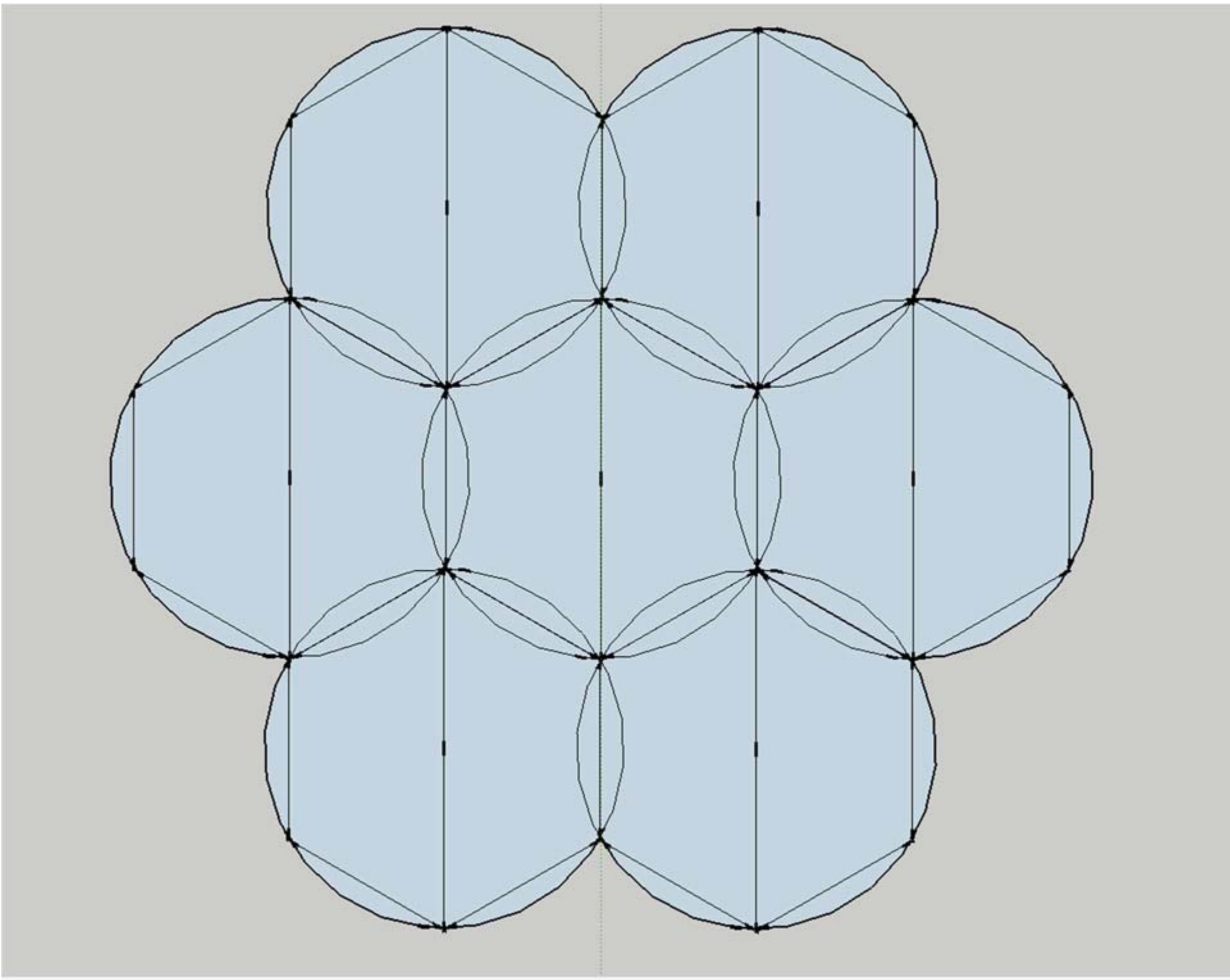
Modelling optimal placement of the balloons



[1] http://www.mapsofworld.com/lat_long/newzealand-lat-long.html
[2] http://www.teara.govt.nz/en/1966/sub-antarctic-islands/page-6
[3] http://www.howderfamily.com/blog/antipodes-islands/
[4] http://www.doc.govt.nz/conservation/land-and-freshwater/offshore-islands/new-zealands-subantarctic-islands/auckland-islands/
[5] http://www.birdlife.org/datazone/ebafactsheet.php?id=431
[6] http://en.wikipedia.org/wiki/Three_Kings_Islands
[7] http://www.doc.govt.nz/conservation/land-and-freshwater/offshore-islands/new-zealands-subantarctic-islands/campbell-island/
[8] http://www.teara.govt.nz/en/1966/kermadec-islands

To calculate the number of balloons we need to get the range each balloon can cover.

This may be done by drawing a hexagon within a circle and then tessellate along with the circle on the outside. This minimises the overlapping area as it ensures that there will be no random small areas which do not have any coverage which require a balloon to provide coverage which will have large amounts of redundancy due to the large overlap with surrounding area.



According to http://www.google.com/loon/how/, they provide a circle of diameter 40km around the balloon with access to the internet. Therefore, based upon this to get the minimum number of balloons we would need to get the circles of which adjacent balloons provide access to minimise their overlapping area. However, as balloons may encounter issues thus no longer providing coverage, we decided to use a model which takes each balloon so that the area which adjacent balloons cover, completely overlap the area which that balloon covers. However, to reduce redundancy we would need to minimise the area of which the area of coverage of adjacent balloons overlap. The best shape that fulfils these conditions are 3 balloons of equal distance from each other on the circumference of the circle which the balloon covers. This means the distance between each balloon will be 20km which we assume it may communicate across as the distance to the edge of the area on the ground to which it provides internet access is approximately 28km via Pythagoras ($\text{Sqrt}(20^2+20^2)$).Thus, we overplayed this on top of a map of New Zealand to scale. Each circle represents the area of which one balloon may provide internet access for

NOTE: Not all circles were counted in the picture on the right

We then, using our modelling program[9], counted the number of circles that overlap New Zealand. This showed the necessity of 5028 being the optimal number of balloons to fully cover New Zealand. This included balloons slightly offshore to accommodate to ensure overlap of all ground area and thus redundancy. It also includes the gap between the north and south islands of New Zealand to accommodate transmission of information between the islands but also for drifting between islands.

[9] See Appendix

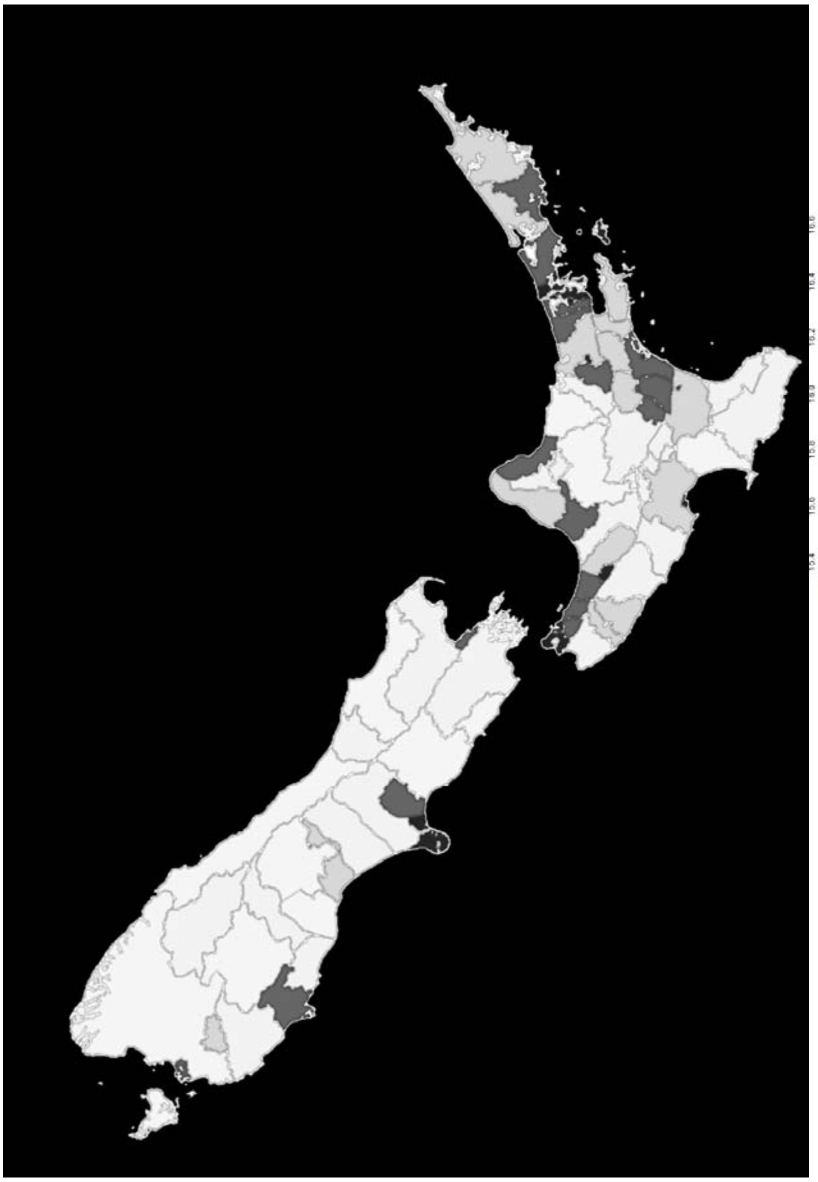
Nature of the model

Our simulation works via a few conditions using a point based system:

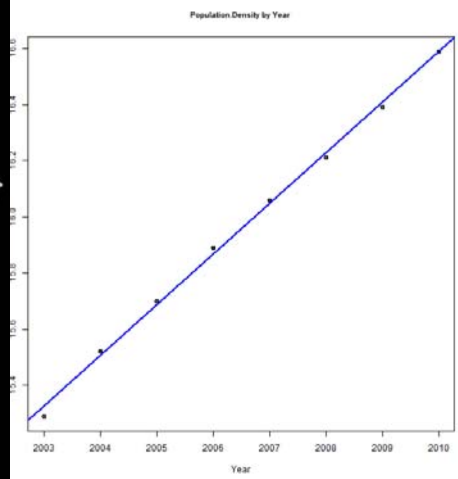
- For each 250m of elevation, 1 point penalty is applied. This is to reflect terrain absorption of wireless signals due to vegetation, line of sight problems, etc.
- For every 50 people living in 1km^2 of land, 1 point penalty is applied. This is to reflect noise and bandwidth requirement. In areas with higher population density, there will be more noise and bandwidth use, which requires more balloons to maintain.
- For every 1 point penalty, the effective range of the signal is reduced by 500m.
- For the minimum coverage test, it is assumed that no balloons move.

Taking in these conditions, the program takes a file with numeral values of size 2200 by 3200 for the altitude. This is translated into points in a 2200x3200 array. [10] The program then takes in a population density map (adapted to fit a more linear scale from NZ Statistics site) and adds further point penalties based upon the population density. Using a gradient map of size 80px by 80px, this is read into an array and then subsequently applies a point advantage at these points. With 40 points at the centre and 1 point at the outer edges of the circle. This reflects the signal of the balloon decreasing by 1 point every 500m. The point of any access point has the penalty added to it. Any blocks with a negative point after this is regarded as having zero coverage and will be marked red. (Note that due to the processing required for the population density map, there are a few red pixels on the generated image. These can be safely ignored).

[10] See Appendix 1



A map is then generated by the program, showing coverage areas as a shade of grey. White=Full Coverage, Black = Little coverage, Red = No Coverage.



Population density map used was also validated through extrapolating. The latest population density census for New Zealand was conducted in 2006 so in order to reflect the current actual population density of New Zealand, a model on iNZight was created to predict the current average population density of New Zealand. The model was plotted with data provided from TradeEconomics.com which provided from 2003 to 2010. This created a linear model which we can see a positive growth in population density through the years. From the summary that was generated, the formula for the relationship between the population density and the year was found to be 0.18036 * Year + -345.93 =Population Density"

By substituting "Year" with 2013, we arrive at the expected population density of 2013 to be 17.13468 per km^2

(Related citation: http://www.tradingeconomics.com/new-zealand/population-density-people-per-sq-km-wb-data.html**)**

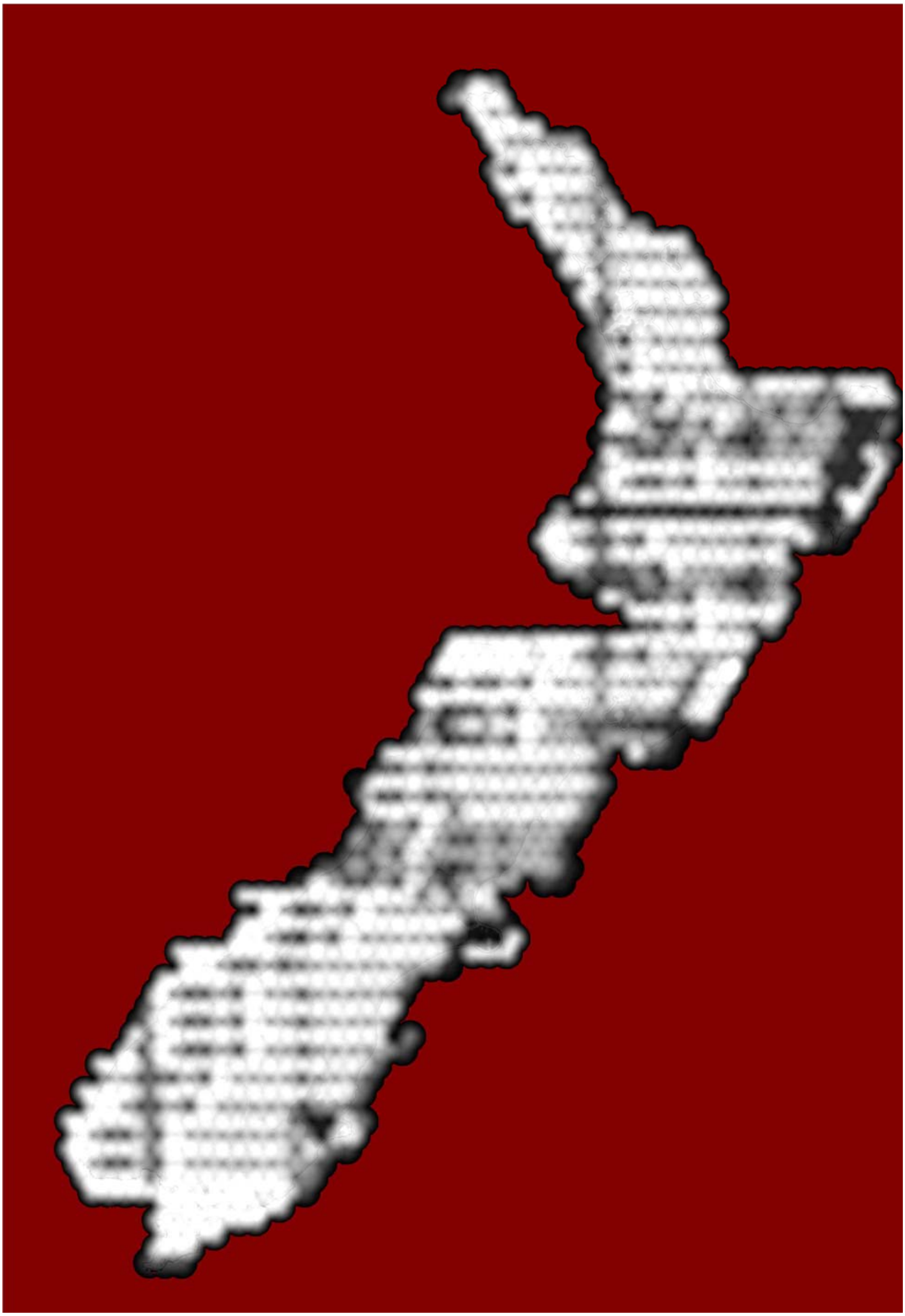
The population density map above has been adjusted for the values created by extrapolating with values from iNZight.

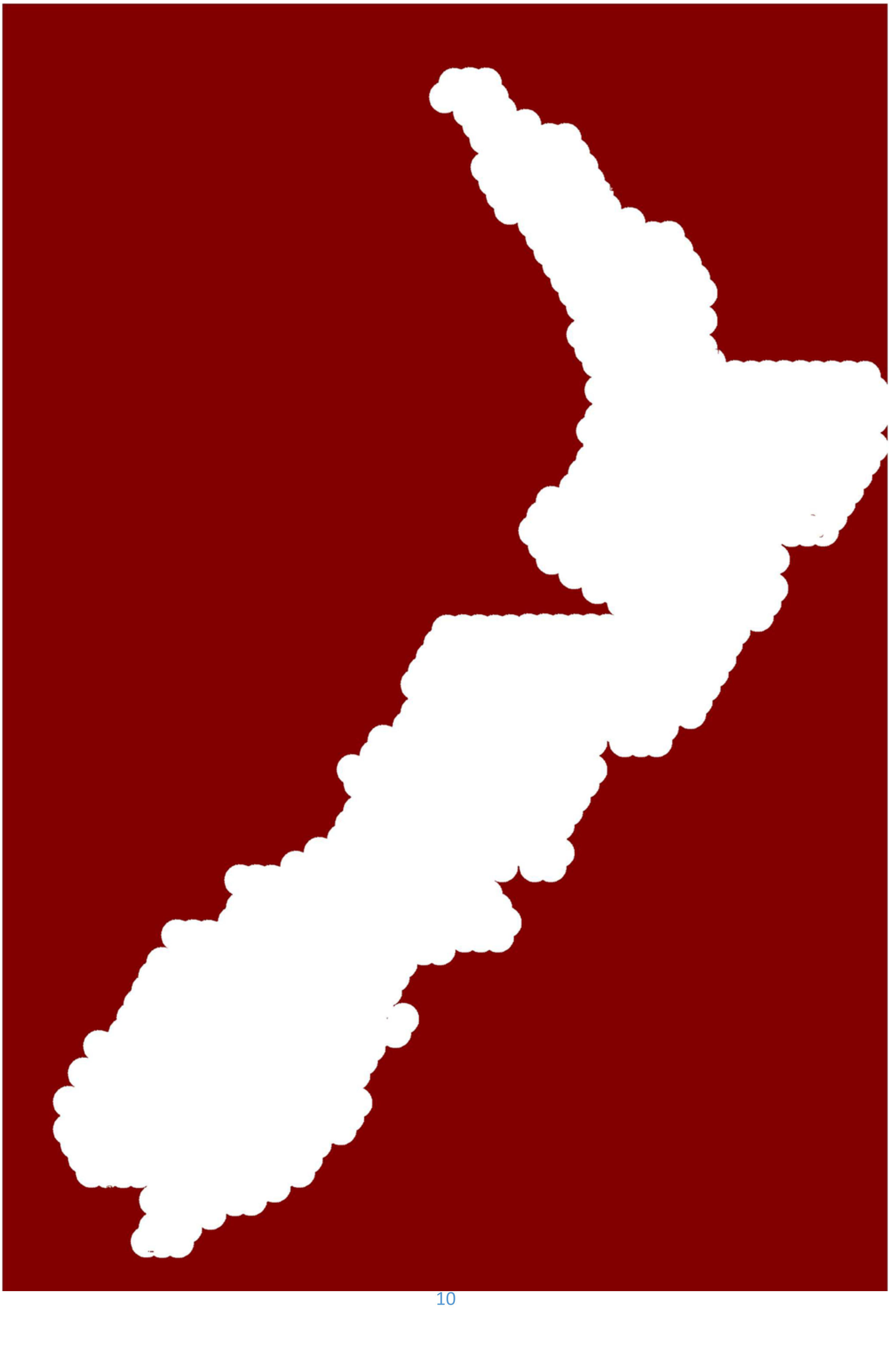
Movement of the balloons

The wind patterns in the stratosphere is predominately following the Westerlies. There are alternate wind paths which the balloons could drift along with, such as circling around Antarctica (travelling along the latitude), crossing over Australia and back to New Zealand or the balloons can circle to South Africa and back to New Zealand.

However, due to Google stating that these balloons are not capable of powered flight, it is assumed that wind in the stratosphere blows in both directions. Due to this, the balloons are capable of changing their course based on the change of altitude alone.

Summary:





This means that if stationary, we only need 5028 balloons to provide access to all of New Zealand. However, in reality the balloons move. According to http://www.google.com/loon/how/, at different altitudes, the wind travels in different directions therefore forcing the balloons to move. The balloons are known to have no means of propulsion and rely on the predictability of the wind in the stratosphere to move.

To accommodate for this we doubled up the number of balloons. To explain the reasoning for this we would need to consider a case where all of the balloons were to move in one direction with the same velocity and once they reach the end they would reverse the direction. If it started with equal distribution across the entire area, then as they travel they would eventually bunch up on one side with half of them moving in one direction and half of them moving in the other direction therefore leaving half of the area without internet access. To avoid this we apply an equal number of balloons above therefore creating a system where balloons move in a circle therefore ensuring the distribution remains the same.

Hence according to this, 10056 balloons is needed. However, in reality the wind patterns will not actually behave according to this. Since we were unable to find sufficient information about wind patterns in the stratosphere we cannot perfectly account for wind in the stratosphere. Therefore, to accommodate for this we must increase the number of balloons for the differences in direction of the wind and differences in speed of the wind. To do so we decided to round up to the nearest thousand which is 11,000 balloons

Appendix 1

Simulation Program written in C# using Visual Studio as the IDE. Note that Main Form requires one button. A copy of this program is attached to this report.

```
using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.IO;


namespace WindowsFormsApplication1

{

    public partial class Form1 : Form
```

```
{
    public Form1()
    {
        InitializeComponent();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        string[] files = new string[5];
        openFileDialog1.ShowDialog();
        string file = openFileDialog1.FileName;
        files[0] = file;
        openFileDialog1.ShowDialog();
        file = openFileDialog1.FileName;
        files[1] = file;
        openFileDialog1.ShowDialog();
        file = openFileDialog1.FileName;
        files[2] = file;
        openFileDialog1.ShowDialog();
        file = openFileDialog1.FileName;
        files[3] = file;
        for (int a = 0; a < 5; a++)
        {
            if (files[a] == null)
            {
                MessageBox.Show("Missing Files");
                break;
            }
        }
    }
```

```
private void openFileDialog1_FileOk(object sender, CancelEventArgs e)
{

}

private void backgroundWorker1_DoWork(object sender, DoWorkEventArgs e)
{
    string[] files = (string[])e.Argument;
    string line;
    int count = 0;
    int[,] simulation = new int[2200, 3200];
    StreamReader read = new StreamReader(files[0]);
    Bitmap pt = new Bitmap(2200, 3200);
    while ((line = read.ReadLine()) != null)
    {
        string[] words = line.Split(' ');
        for (int a = 0; a <= words.GetUpperBound(0) - 1; a++)
        {
            int x = Convert.ToInt32(words[a]);
            if (x > 0)
            {
                int point = (x / 250);
                simulation[a, count] = -point;
            }
        }
        count++;
    }
    Bitmap bitmap = new Bitmap(files[1]);
    for (int a = 0; a < 2200; a++)
    {
        for (int b = 0; b < 3200; b++)
```

```
        {
            Color colour = bitmap.GetPixel(a, b);
            int point = (Convert.ToInt32(colour.R.ToString()));
            if (point > 0)
            {
                if (point < 247)
                {
                    float points = ((247 - point) * 4) / 50;
                    point = Convert.ToInt32(points);
                    simulation[a, b] = simulation[a, b] - point;
                }
            }
        }
    }
    Bitmap coverage = new Bitmap(files[2]);
    int[,] cover = new int[80, 80];
    for (int a = 0; a < 80; a++)
    {
        for (int b = 0; b < 80; b++)
        {
            Color colour = coverage.GetPixel(a, b);
            cover[a, b] = (Convert.ToInt32(colour.R.ToString()));
        }
    }
    Bitmap accesspoints = new Bitmap(files[3]);
    int[,] ap = new int[2200, 3200];
    for (int a = 0; a < 2200; a++)
    {
        for (int b = 0; b < 3200; b++)
        {
            Color colour = accesspoints.GetPixel(a, b);
```

```
            if (0 == (Convert.ToInt32(colour.R.ToString())))
            {
                ap[a, b] = 1;
            }
        }
    }
    for (int a = 0; a < 2200; a++)
    {
        for (int b = 0; b < 3200; b++)
        {
            if (0 == (ap[a, b]))
            {
                for (int c = a - 40; c < a + 39; c++)
                {
                    for (int d = b - 40; d < b + 39; d++)
                    {
                        if (c < 0)
                        {
                            c = 0;
                        }
                        if (d < 0)
                        {
                            d = 0;
                        } if (c > 2199)
                        {
                            continue;
                        }
                        if (d > 3199)
                        {
                            continue;
                        }
```

```
                                simulation[c, d] = simulation[c, d] + cover[(c + 40 - a), (d
+ 40 - b)];
                            }
                        }
                    }
                }
            }
            for (int a = 0; a < 2200; a++)
            {
                for (int b = 0; b < 3200; b++)
                {
                    if (simulation[a, b] > 0)
                    {
                        pt.SetPixel(a, b, Color.FromArgb(255, 255, 255));
                    }
                    if (simulation[a, b] < 1)
                    {
                        pt.SetPixel(a, b, Color.FromArgb(0, 0, 0));
                    }
                }
            }
            pt.Save("map1.png");
            MessageBox.Show("Done");
        }
    }
}
```