

# Geolocation based on signal strength values

## Motivation

**dc-geolocation** is a program that allows a wireless networking device to find its own location using the information it can gather from its network access points. This can be any device that is capable of performing wireless networking ie. mobile phones and laptops.

Self-location can be used for several important tasks. These tasks include locating someone in an emergency, navigation within small areas (think of something like GPS for an apartment building), or tracking of that device.

The only data self-location needs is the location of 4 known points along with their distance to the unknown point. From there, what's necessary is the mathematical framework to convert that data into our physical location. To collect this data we obtain the information using wireless network access points (AP's).

This project has produced two things

- A program that collects the "signal strength" from each access point, also known as the RSSI value.
- The mathematical framework that turns the needed data into our location through the process of multilateration, with a python program to accompany it.

dc-geolocation is written by David Chamberlain [catmanisacatlord@gmail.com](mailto:catmanisacatlord@gmail.com) of Santa Fe High School and the Institute for Computing in Research.

## Accessing information in a wireless access point

Most people are only familiar with their access point from the web-based administrative interface that they use to set it up, but one can also use instructions to automatically get specific information from them.

To get a feeling for how this works, try the following at the command line of a GNU/Linux system:

```
sudo iw dev wlp1s0 link
```

This command will give you some basic information about the wireless network you are currently connected to, while While this command will give you all of the information about all of the networks in your area:

```
sudo iw dev wlp1s0 scan
```

We use the second command in our program due to this functionality and then filter away the networks and data we do not want. This leaves us with the RSSI value along with the SSID for the AP's that we desire. Using the Free-space path loss formula (which determines how a signal gets quieter as it passes through

free space) and rearranging it, we can achieve a distance in meters using the formula below.

Unfortunately, a big problem with using the signal strength with routers is how messy and scattered the signal can be, if you have that 1 wifi dead zone in your house you will understand what I mean. To get the most accurate data it's best to use it in situations where direct LOS(line of sight) is established between the AP and whichever wireless device is being tracked. Ontop of this, using a 5200 MHz frequency can hopefully reduce any noise.

## The mathematics of geolocation

In order to calculate the unknown position of a point (ex, ey, ez), we must have 4 known points with known locations in 3D space (an\_x, an\_y, an\_z) along with their distance to our unknown point (e\_a1,e\_a2,e\_a3,e\_a4). From there we estimate the point (ex,ey,ez) so that the difference in the distance from the estimated point to each AP and the measured distance is minimized accordin to the least means square algorithm.

In my setup I placed each of my 4 routers on each vertex 1 meter square with my laptop in the middle as the wireless device. This shape was chosen due to its ease in marking the location of the AP's. The diagram below depicts this, with the black circles representing AP's and the red circle depicting the laptop. All units are in meters.

## How to run the software

1. Download and install python 3.8.
2. Extract the name.zip file to a directory of your choosing.
3. From your terminal run `pip install matplotlib numpy pandas` . This will install the necessary python packages.  
  
Next, make sure that the command `sudo iw dev wlp1s0 scan` can run without needing a password.
4. In a new terminal run `sudo visudo` . This will allow you to edit your sudoers file (the file that delegates permissions for your device) using the nano text editor.
5. Navigate to the end of the text file and type in `Username ALL=NOPASSWD: /sbin/iw` , for example, `david ALL=NOPASSWD: /sbin/iw`
6. Press `ctrl+x` to exit the file, next press `y` to save the modified buffer, then press `enter`.

NOTE: This should bring you back to your terminal. If the terminal says you have any syntax errors, press `x`. This will cancel all the changes you made to the sudoers file. You will need to restart the process at step 4.

7. Open up `RSSIVvalues.py` in the editor of your choice and edit the `NET-WORKS` variable to the `SSID` values (the name of it) of the AP's (access points aka routers) you have.
8. Set the `FREQUENCY` variable to the bandwidth you are using for your AP's to communicate.
9. Now you finally run the program in a terminal using “`python3 pathToFile/RSSIVvalues.py`”