Table of Figures

[Figure 3.1: Raspberry pi 2](#_Toc40869841)

**3.methodology**

In this section, the methodology and the main features of the devices employed to perform this study is described. In this implementation, An attempt was made to find out what would be the best device configured as a access point for extending Wifi network while attached to a drone For this to happen, the device needed to be small and as lightweight as possible so it would be able to be fit on the drone.

The Raspberry Pi was used as the central element of the system. This inexpensive single-board computer offers to perform this task. Thus, this device is able to deploy a wireless network acting as network nodes within the system architecture and allowing the end-users/things to gain connectivity through them. WIFI is so far the most widespread access network for providing connectivity to end-users’ wireless devices. This development board feature a [Broadcom](https://en.wikipedia.org/wiki/Broadcom) [system on a chip](https://en.wikipedia.org/wiki/System_on_a_chip) (SoC) with an integrated [ARM](https://en.wikipedia.org/wiki/ARM_architecture)-compatible [central processing unit](https://en.wikipedia.org/wiki/Central_processing_unit) (CPU) and [on-chip graphics processing unit](https://en.wikipedia.org/wiki/Graphics_processing_unit#Integrated_graphics) (GPU)**(**Raspberrypi.org, 2020). Specifically, the raspberry pi 3 was used in the experiments conducted. This board ran the latest version of Raspbian (Raspbian Buster 4.19). Regarding the Raspberry pi board power supply, an external battery of 10400 mAh was used. This battery allows up to 5 hours of RPI’s working time, depending on the operational mode and the traffic load(Pi?, W, 2020). More information about the lifetime will be provided in the results section. The BCM43438 wireless LAN which comes integrated on the raspberry pi 3 model was used. This network card allows for connections up to 38.8Mbits/s, supporting several Wi-Fi standard versions, namely, IEEE 802.11a/b/g/ad. This wireless card can operate as an access point, or as an ad-hoc node within a mesh network. In addition, i also used a Vodafone 3g usb WIFI dongle connected directly to the raspberry pi board to obtain an internet connection which than was transmitted through the raspberry pi in built WIFI providing large coverage range. The total weight of the board, the battery, and the WIFI dongle is approximately 300g.

Figure 3.1: Raspberry pi

The work was divided into three phases. Firstly, the installation of the necessary software

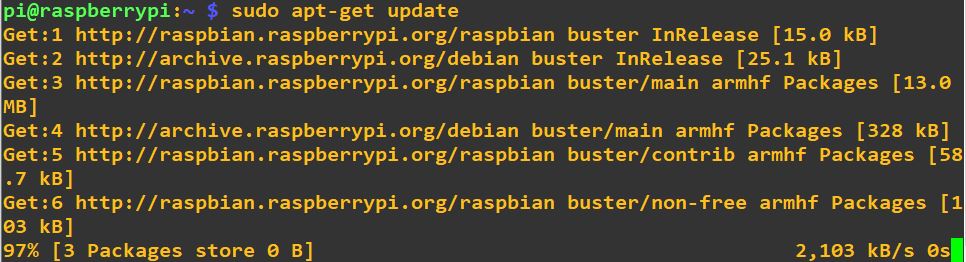
to be able to configure the raspberry Pi board to act as a WiFi node was done, The Raspberry Pi will need to have access point software installed, along with DHCP server software to provide connecting devices with a network address. Second, the setup was to set up the raspberry Pi as a wireless access point. This is done with the use of the software that was installed in the above phase.

lastly, The performance of the raspberry Pi board was experimentally tested as an intermediate network within a WiFi network. These tests were performed first in a controlled environment (static on-the-ground) assuming an open-air scenario and then where it was in a real aerial deployment with a drone. In both cases, the two main operational frequencies of the IEEE 802.11 were studied.The usual mode of operation is the 2.4GHz, which is implemented in most commercial WiFi routers the 2.4 GHz band is great for connecting over longer distances but offers slower speeds due to more traffic on the network. the 2.4 GHz band also cannot support as many devices and can quickly become bogged down.

On the other hand, the 5 GHz band offers coverage for shorter distances but provides the users with faster speeds. While the 5 GHz band is newer and offers more channels compared to the 2.4 GHz band, it can’t reach as far since higher frequencies cannot penetrate solid objects, such as walls and floors. However, having higher frequencies allow the data to be transmitted faster than lower frequencies, so the 5 GHz band allows you to upload and download files faster. Based on these models, we calculated the maximum expected uplink and downlinkradio coverage for several versions of the WiFi bands.

**Installations**

The first step was to install all the necessary software on the raspberry pi, in order to be able to configure the device. The code necessary to turn the Raspberry pi to a router was all applied in the terminal windows on the Raspbian software which is equivalent to Command Prompt that is found on windows.

Before installing several software on the device the following code was used beforehand: ‘**sudo apt-get update’ and** ‘**sudo apt-get upgrade’**. Both lines of code were used in the setup to update the device to make sure the software is latest versions to always use the latest versions of applications. The first keyword in the lines of codes above, which will also be seen in most of the other terminal commands and screenshots, is the word **“sudo**”. This command means “superuser do” which provides an efficient way to give users administrative permission to use specific system commands.

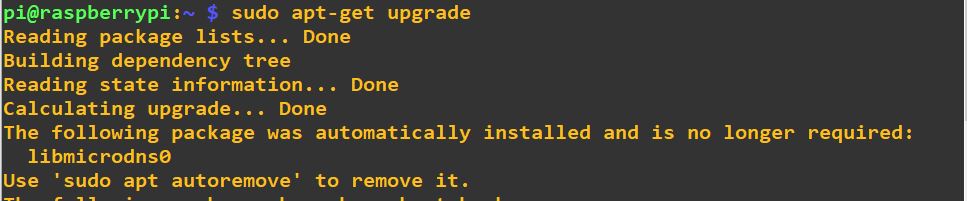
 Figure 3.2: Updating RPi

Figure 3.3: Upgrading RPI

Firstly, two programmes called “hostapd” and “bridge-utils” were needed to be installed on the Raspberry Pi, these two programs are necessary to be able to make your Raspberry Pi into a wireless access point.

Then, the program that was just installed will be temporary stopped since the programs configuration files will be edited so to be safe, they were turned off before starting to make necessary changes.

**The setup**

After having all the required software installed it was time to start setting up the config the first step was to deny the interfaces that will be used for receiving and providing the network from being automatically allocated an IP addresses by the DHCP client of the Raspberry Pi. This was done by going in the config file dhcpcd.conf using the following command “**sudo nano /etc/dhcpcd.conf**” and writing “denyinterfaces wlan0” and “denyinterfaces eth0” at the bottom of the config file.

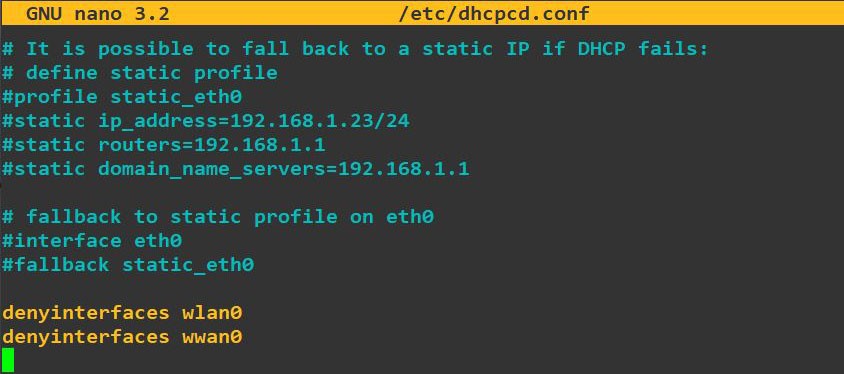
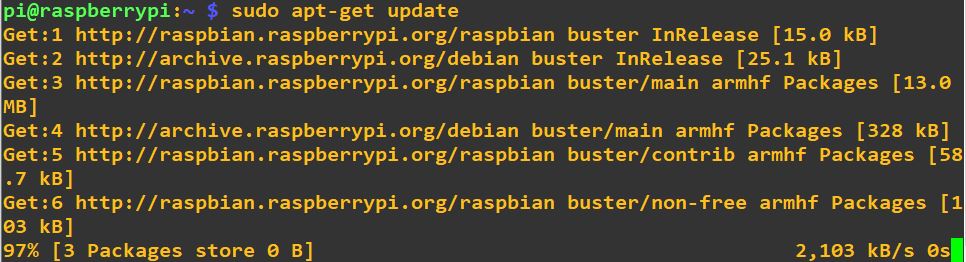


Figure 2: Dhcpcd.conf file

Next a bridge was created “**sudo brctl addbr br0**” which in this case was called br0. This feature allows to ability to connect two network interfaces together. To do so the interfaces file needs to be edited to adjust the various devices to work with bridging. To make this work with the newer systemd configuration options, a set of network configuration files must be created.



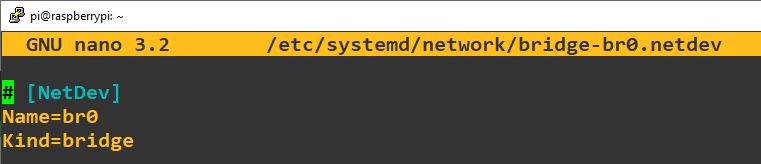
In the network file “bridge-br0.netdev”, the created br0 virtual network needs to be defined as a bridge, this was done by configuring the interface by writing “Name-br0” and “Kind=bridge”.

Figure 3.3: Bridge-br0.netdev file

Right after defining the bridge, it was assigned an IP address of 192.168.10.100/24, A bridge does generally does not *require* an IP address. There are many situations where there is no need to have one. However, there are some cases where it is possible to have one such as: When the "primary" Network card is a member of the bridge, such that the bridge is your connectivity to the outside world. In this case, rather than assigning an IP address to (for example) wlan0, you would assign it to the bridge device instead

Figure 3. 4: bridge network

Finally the systemd-networkd program can be restarted to have the bridge network running with the applied settings.

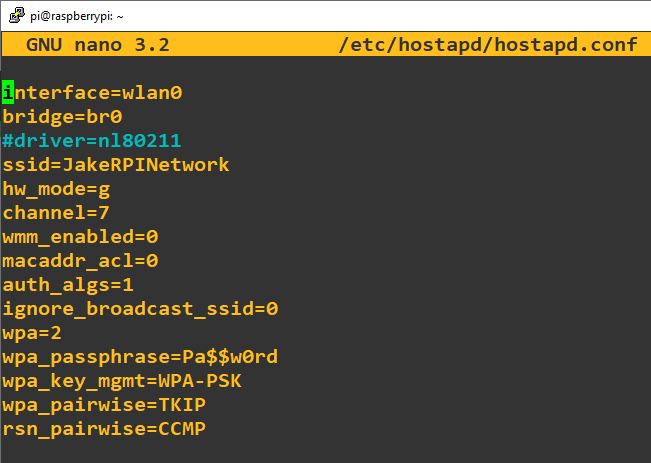
The final step to this setup was to add the various parameters for your wireless network into the hostapd configuration file, located at /etc/hostapd/hostapd.conf as this was a new/empty file. Hostapd (Host access point daemon) is a user space software access point that is capable of turning normal network interface cards into access points and authentication servers.There are many, many other options which can be set in this configuration file, but these were the ones needed for this setup. This configuration assumes we are using channel 7, with a network name of JakeRPINetwork, and a password Pa$$w0rd.

Figure 5:2.4Ghz hostapd.conf file

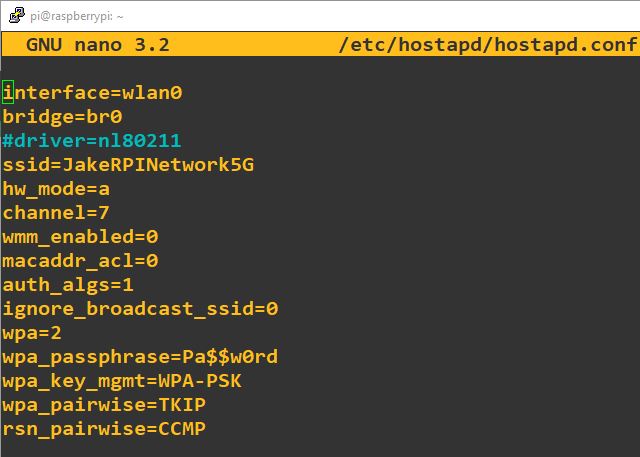


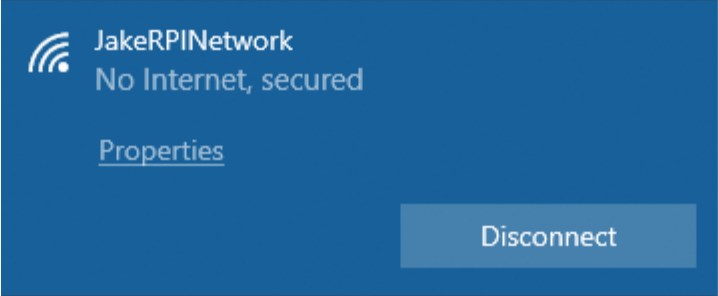
Figure 3.6:5Ghz hostapd.conf file

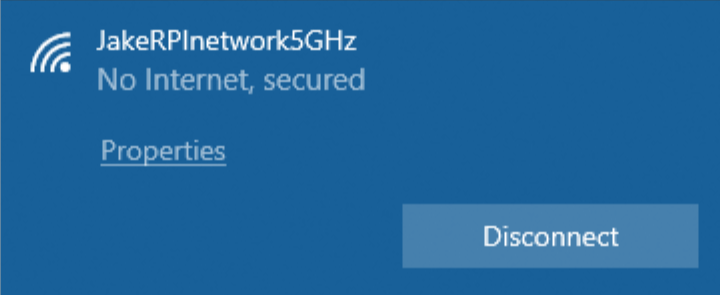
To set up the 5GHz network it was an identical setup to the 2.4Ghz network but with minor changes to the hostapd.conf file. To use the 5 GHz band, what was required is to change the operations mode from hw\_mode=g(IEEE 802.11g) to hw\_mode=a(IEEE 802.11a). The ssid name was also changed to identify which network is the 5Ghz and 2.4Ghz while testing performing tests.

At this point all the setup is complete but it is still not working, this is due to turning the service off at the start of the installation. To enable the hostpad service the following commands had to be used: “**sudo systemctl unmask hostapd” , “sudo systemctl enable hostapd”** and “**sudo systemctl start hostapd**”.There should now be a functioning bridge between the wireless LAN and the 3g dongle connection on the Raspberry Pi, and any device associated with the Raspberry Pi access point will act as if it is connected to the access point's wired Ethernet. The bridge will have been allocated an IP address via the wired Ethernet's DHCP server

**Testing**

The first network test that was conducted was trying to connect with the networks created which very simple to do. However, although connecting to the network successfully there was no internet connection due to not having a sim card with a 3g plan in the USB 3G dongle. For the testing that will be done there was no need to have internet access so this will not affect the results that will be obtained.





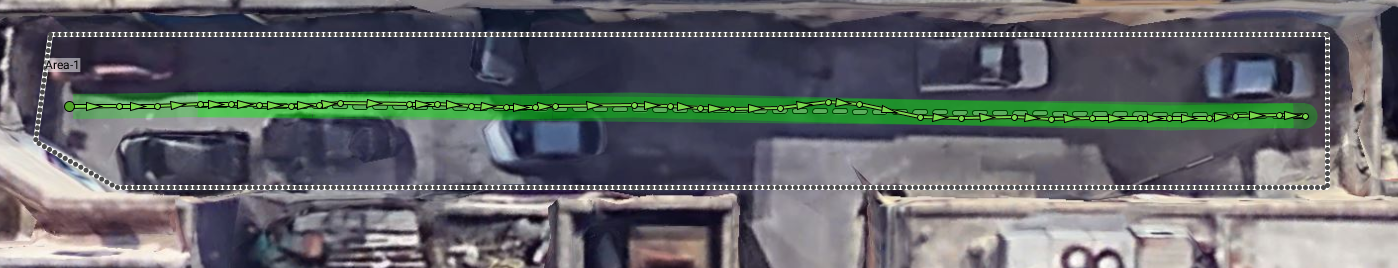
When the two network connections were both able to be connected to, this meant that the setup had worked and was ready for the testing phase where they will be compared with each other to find out which would be most suitable for extending wireless networks while equipped on a drone.

Since the connection was successful, a software was installed to be used for testing the networks. The name of the software is Ekahau Site Survey, which is created by Ekahau, Inc (2000). Ekahau Pro/Site survey is a site survey tool that shows where to place and how to configure your access points by accurately predicts network coverage, performance and capacity of the network. In this setup it was not used to identify the suitable location for the access point but used to identify the network performance.

After opening the software, the first step was to inset a map of the location I had done the real-world tests. An image using google maps of the location was inserted into the software and then scaled to the correct distance. The reason for this was to be able to measure the distance from any point on the map to the location of the access point. After an outline of the road was done, this was not required but I focused on places which were reachable by walk during this implementation. Lastly it was time to start the survey. While being connected to the Access point network a continues survey was taken. This is done my walking around the facility with the laptop and clicking on the map the current location. This was done until all ground was covered the survey was finished. In the below figures are example of how this was implemented in the software.







References

Raspberrypi.org. 2020. [online] Available at: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/>.

Pi?, W., 2020. *Will Any External Battery Power A Raspberry Pi?*. [online] Raspberry Pi Stack Exchange. Available at: <https://raspberrypi.stackexchange.com/questions/26705/will-any-external-battery-power-a-raspberry-pi>.