Article title

Building an Open-Source End-to-End Cellular Network using Software-Defined Radios and UAVs

Authors
Keith Powell, Andrew Yingst, Vuk Marojevic

Affiliations
Mississippi State University
Mississippi State, MS 39762

Corresponding author's email address and Twitter handle
kp1747@msstate.edu

Abstract
The uses for unmanned aerial vehicles (UAVs) are expanding quickly but effective communication beyond traditional radio line of sight is needed. Cellular networks are a promising solution but were designed for ground-based users. Aerial users introduce a host of questions that must be answered by researchers. Software-defined radios (SDRs) provide a relatively inexpensive entry point to working with cellular networks. Open-source software solutions bring cellular networks to SDRs with strong communities offering support and contributions to the projects and users. This article introduces an experimental setup to investigate such problems with clear instructions to design, build, and operate the platform.

Keywords
Wireless Communication; Unmanned Aerial Vehicle; Open-Source Software; Software-Defined Radio;

Table 1. Specifications

Hardware name	SDR UAV Cellular Network
Subject area	Educational tools and open-source alternatives to existing infrastructure
Hardware type	Electrical engineering and computer science
Closest commercial analog	Ukama
Open source license	CC0 1.0 Universal
Cost of hardware	US$22000
Source file repository	https://www.doi.org/10.17605/OSF.IO/R3KV7

1. Hardware in context

There are a number of tools that allow users to create their own cellular networks with variable feature sets. However, at the time of writing, there is not an all-in-one solution where users can deploy a cellular network with the user equipment (UE) or part of the network infrastructure located on a UAV. Many commercial products exist allowing users to create an eNodeB (eNB), the radio base station of a cellular network, that serves as a provider or can connect to an existing provider such as Blinq Networks [1], Airspan [2], or the crowdsourced Ukama [3]. These solutions are aimed to supply cell service to an existing user equipment (UE) such as a personal cell phone. Additionally, these products all operate in the Citizens Broadband Radio Service (CBRS) band, a shared band that is free to use under certain stipulations. The system discussed in this work operates in long-term evolution (LTE) band 22, but is fully capable of operating in the CBRS band as well. Unlike commercial solutions, this system features several parts to create each of the network nodes rather than a single unit that handles all functionality. The core network and eNB are collocated as a ground node. The UE is located on a portable aerial node, carried by a UAV. The roles of the two nodes can be swapped and the network can be directly scaled by adding more eNBs and UEs with a wired or wireless backhaul, depending on the configuration. Software-defined radios (SDRs) are used on both nodes to transmit and receive the LTE signals. Commercial solutions include options for both local coverage using in-home devices or widespread coverage using outdoor devices such as pole-mounted systems. In many cases, it is expected that these devices will be connected to some external network via an internet service provider. The presented design is also capable of extending an external service such as internet.

2. Hardware description

2.1. System Overview

This system is based around using SDRs to create a cellular link between two nodes. In this case, the nodes have different sets of constraints as one is stationary on the ground and the second is mobile in the air. Therefore, many components vary between the two nodes. The SDRs used are Universal Software Radio Peripherals (USRPs) from Ettus Research. To communicate with the PCs, these require the USRP Hardware Driver (UHD). Additionally, two software suites are used such that the SDRs can create the cellular link, srsRAN and OpenAirInterface (OAI). Both srsRAN and OAI are free and open-source while supporting both 4G and 5G radio links. A spectrum analyzer is collocated with the ground node to provide additional measurements and monitor the spectrum that is being used. The WiFi router is located separately and used to control the software on the aerial node PC remotely from the ground node. Figure 1 shows an overview of the system and its major subsections. The system is able to handle up to a 20 MHz LTE connection with uplink and downlink throughputs to the LTE specification. The configuration of the experiment described here operates around 3.5 GHz.

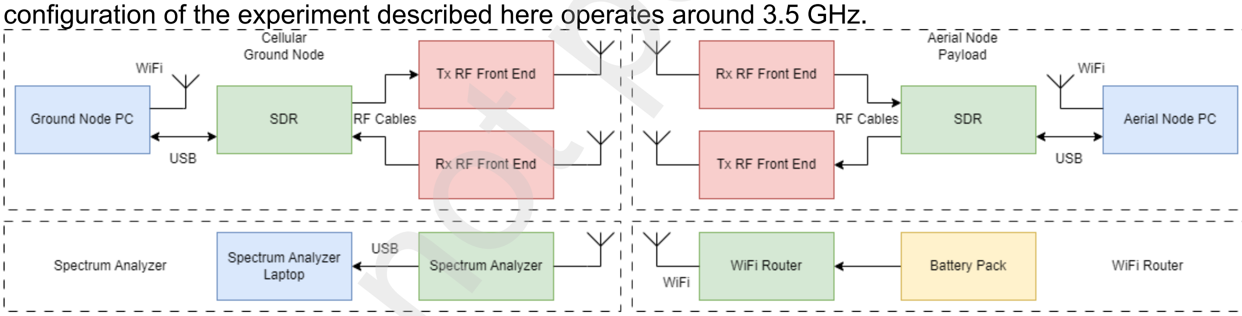

Fig. 1. Overview of SDR UAV Cellular Network design. The ground node (left) consists of the cellular SDR chain (top) and spectrum analyzer chain (bottom). The aerial node (top-right) shows the payload consisting of the UE cellular SDR chain. The WiFi Router (bottom-right) is located on the ground, but separate from the ground node.

2.2. Ground/Aerial Node PCs

The ground and aerial node PCs have similar constraints as both are required to power and operate the SDRs while maintaining the full capacity of the LTE network. Both PCs should have a multi-core i7 or i9 processor that can handle at least 3 GHz consistently. This is required for srsRAN or OAI to operate in real-time, especially at larger bandwidths. Additionally, due to the high bandwidth communication between the PC and SDR and the SDR's power requirements, each PC must be equipped with USB 3.0. This system uses the B205 mini-i USRP that utilizes micro-USB 3.0. In order to control the aerial node PC while it is flying, WiFi is used to access the mobile node computer via secure shell (SSH) wirelessly. The ground node PC is safely able to use ethernet or WiFi connected to the WiFi router based on available tools.
The ground node PC used for this system is a Dell small form factor featuring an Intel i9-9900 processor. This PC does not have any power requirements due to readily available power on the ground. The small form factor is convenient for transporting between experiments, but is not required for the ground node. In order to maximize the range for communicating with the aerial node PC, the ground node PC is equipped

with a USB WiFi adapter instead of ethernet cable to cover the distance from PC to WiFi router. The aerial node PC is an Intel NUC 10 featuring an i7-10710U processor. The NUC 10 is ideal for an aerial node as it is a mini-PC and is reasonably lightweight. The i7-10710U processor is also a low power processor that satisfies the required performance for the system while minimizing the amount of power consumed. Weight and power consumption are both important to consider for the battery powered aerial node to ensure performance is maintained and the UAV is able to fly safely for as long as possible. This PC also comes preinstalled with a WiFi 6 card, and it is installed under the UAV payload tray, upside down for best WiFi signal strength. Both PCs utilize the same software packages that are discussed later.

2.3. SDRs

The SDRs used for both the aerial and ground nodes is the USRP B205 mini-i. This SDR operates using a micro-USB 3.0 cable connected to a companion computer for power and data transfer. On the ground node, this is less necessary as power supply and consumption are less constrained than the aerial node. On the aerial node, this SDR is ideal as is has a low power consumption compared to other available SDRs and does not require an external power supply for any operation it may perform. Additionally, the "-i" model is used as it includes an industrial grade Spartan-6 FPGA that is tested to perform at a wider range of temperatures, ideal for the aerial node where the device is exposed to a wider variety of environmental conditions. The enclosure for this SDR is also utilized due to these conditions. The B205mini-i also includes an AD9364 radio-frequency integrated circuit transceiver that is capable of 56 MHz of instantaneous bandwidth at a desired frequency between 70 MHz and 6 GHz. This works well for LTE experiments as the maximum bandwidth used is 20 MHz with all standard LTE bands existing between 410 MHz (band 87) and 5925 MHz (band 47).

2.4. RF Front Ends

Both the transmit and receive RF front ends consist of three major parts: an amplifier, filter, and antennas. On the ground node, power consumption and size are not of concern. Therefore, the Mini-Circuits ZHL-15W-422-S+ power amplifier (PA) is suitable to maximize the transmit signal. This is a 15 W amplifier allowing 46 dB of gain in the ranges of 600 MHz to 4200 MHz with SMA connectors to match the SDRs and other components. Due to the high power of this amplifier, it is important to follow the given instructions to prevent damage or undesired functionality when connecting it to the system. Additionally, the output of this chain should be monitored to ensure the output signal is not distorted or causing excessive out-of-band emission called spectral regrowth. On the UAV, the smaller, lightweight, and less powerful Mini-Circuits ZVE-8G+ PA is selected for the transmit chain which can be used without a heatsink. This is a 1 W PA that has a 34-36 dB gain and operates in the ranges of 2-8 GHz. Similar precautions to the ground node amplifier should be taken to ensure proper operation and appropriate transmissions occur. The receive chain for both the ground and aerial nodes uses the Mini-Circuits ZX60-83LN12+ low noise amplifier (LNA). This PA is very small and lightweight operating between 500 MHz and 8000 MHz. It offers a 22 dB gain to increase the received signal power within the channel selection filter, or preselector, and improve the overall range of the experiments.

The filters on the front ends are used to eliminate harmonics in the transmitter and minimize the out-of-band interference and noise entering the receiver. In other words, the filter in the transmit path is for avoiding spurious transmission at multiples of the carrier frequency, whereas the filter in the receive path acts as a preselector. This allows for a more effective experiment to be performed while preventing interference to outside devices. Both transmit chains use the Mini-Circuits VLF-4400+ low pass filter allowing frequencies from DC to 4400 MHz. The reason for choosing this filter was the original plan of AERPAW, Aerial Experimentation and Research Platform for Advanced Wireless, to offer experiments at 2.5 and 3.3-4.2 GHz where only the bandpass filters in the receive chain needed to be swapped. The receive chains both use the Mini-Circuits VBFZ-3590+ bandpass filter allowing frequencies from 3000 MHz to 4300 MHz. These commercial-off-the-shelf filters are used because of the wide availability and for the desired frequency agility. Commercial cellular networks and UEs use custom and much narrower surface acoustic wave filters for better communications performance in crowded spectrum.

The antennas are selected based on their wide frequency range, maximum power output, omnidirectionality for 3D coverage, and suitability for each node. On the ground node, both transmit and receive chains utilize the Mobile Mark RM-WB1-DN-B1K antenna. This antenna has ranges from 617 MHz to 960 MHz and 1700 MHz to 6000 MHz allowing up to 10 W. This antenna is ideal for the high-power station on the ground but larger and heavier than is appropriate for the aerial node. Instead, the aerial node features the Octane Wireless SA-1400-5900 antenna with ranges from 1400 MHz to 5900 MHz and up to 500mW. This antenna sacrifices the coverage range of the ground antenna in favor of a significant reduction in size and weight. The RF front end components are connected to each other and the SDR using short and low-loss SMA cables.

2.5. Spectrum Analyzer

The spectrum analyzer is a useful tool for troubleshooting and verifying signal outputs from any of the nodes. It is highly recommended that this tool is used to monitor experiments. However, it is not required in order to perform the experiment itself. The spectrum analyzer used in this experiment is the RSA306B. The RSA306B has a range of 9kHz to 6.2 GHz with up to 40 MHz of capture bandwidth. It is connected to a Lenovo Thinkpad with a 4-core Intel i5 8250U that runs the Microsoft Windows operating system. The software required to run this device is only available on Windows. Similar to the SDRs, this spectrum analyzer requires USB 3.0 to operate at its full capacity. A wideband directional antenna, the RFSpace TSA600-6000, is selected allowing for a wide range of frequencies to be captured. The spectrum analyzer and antenna have different RF ports and require an adapter to connect to each other. The directional nature of the antenna also allows for the location of signals to be identified more easily, such as the transmit signal coming from a distant aerial node. The spectrum analyzer is used before the experiment to identify any potential transmissions that are taking place near frequencies that will be used for the experiment in the experiment area. The FCC Website needs to be consulted for any registered active or passive RF users or scheduled experiments. Once it is verified that the area is free of active transmissions, a test run of the experiment is performed with the aerial node still on the ground over a short line of sight distance between the eNB and UE of 5-15 m. The spectrum analyzer is used to check that both SDRs are transmitting clean signals that do not cause significant out of band emission beyond the experimental frequency range. After the test run confirms that all nodes are operating as expected, the analyzer will continue to be used during the experiment to monitor the signals and collect supplementary data to the main experiment's data.

2.6. WiFi Backhaul and Control Channel

The final node of the experiment is the WiFi backhaul to enable deploying multiple eNBs in a field without fiber backhaul access and to facilitate experiments with aerial base stations. The WiFi link is also used to access all computers involved in the experimental SDR network--fixed and mobile--via ssh and all times for controlling the experiment. The reason for choosing WiFi 6 is that it supports multi-user multiple-input, multiple-output communications, for supporting multiple simultaneous links of high data rates and extended ranges. The IEEE 802.11ax is also readily supported by the Intel NUC10 companion computer for the SDR on the UAV and WiFi 6 chips are widely available for PCs and are inserted into the Dell small form factor PC of the fixed SDR node. This node consists of a WiFi 6 router that is collocated with an Anker PowerHouse 200 battery pack. The router is configured to operate at a specific channel in the high 5 GHz frequency range. This channel is further away from the experimental frequency than the alternative 2.4 GHz option, reducing the likelihood that WiFi, cellular, and UAV remote control communications interfere with one another, limiting their capacity. Additionally, the spectrum analyzer is used to confirm that the selected channel is free of interference before operation. Although the band is not regulated, it is essential that the WiFi connection is held between the nodes to maintain proper control of the aerial node computer. The battery pack for this system allows for the router to be placed at a remote location away from the ground node. By taking advantage of the WiFi range for both nodes, longer range experiments can be performed.

3. Design files summary

Table 1. List of files

Design file name	File type	Open source license	Location of the file
enb.conf	Configuration file	CC0 1.0 Universal	https://www.doi.org/10.17605/OSF.IO/R3KV7

epc.conf	Configuration file	CC0 1.0 Universal	https://www.doi.org/10.17605/OSF.IO/R3KV7
mbms.conf	Configuration file	CC0 1.0 Universal	https://www.doi.org/10.17605/OSF.IO/R3KV7
rb.conf	Configuration file	CC0 1.0 Universal	https://www.doi.org/10.17605/OSF.IO/R3KV7
rr.conf	Configuration file	CC0 1.0 Universal	https://www.doi.org/10.17605/OSF.IO/R3KV7
sib.conf	Configuration file	CC0 1.0 Universal	https://www.doi.org/10.17605/OSF.IO/R3KV7
ue.conf	Configuration file	CC0 1.0 Universal	https://www.doi.org/10.17605/OSF.IO/R3KV7
user_db.csv	Database file	CC0 1.0 Universal	https://www.doi.org/10.17605/OSF.IO/R3KV7
enb.band7.tm1. 100PRB.usrpb210.conf	Configuration file	CC0 1.0 Universal	https://www.doi.org/10.17605/OSF.IO/R3KV7

File Description

enb.conf – Configuration file for srsENB. This includes settings for communicating with the core network, RF parameters, and scheduling options.

epc.conf – Configuration file for srsEPC. This includes settings for binded addresses and other core network identifiers.

mbms.conf – Configuration file for srsMBMS.

rb.conf – Configuration file for srsENB. This includes settings for various timing values used.

rr.conf – Configuration file for srsENB. This includes settings for eNB identifiers, handover, and timing.

sib.conf – Configuration file for srsENB. This includes settings for the system information blocks.

ue.conf – Configuration file for srsUE. This includes settings for RF parameters and user identifiers.

user_db.csv – Database file for srsEPC. This includes information on users that are allowed to connect to the network.

enb.band7.tm1.100PRB.usrpb210.conf – Configuration file for OAI. This includes settings for RF parameters, communicating with the core network, and channel parameters.

4. Bill of materials summary

Table 2 presents a subset of the BOM, which captures the major cost-driving components. Individual electrical components are specified on the full BOM but many small mechanical items, like fasteners, are left to the builder's discretion and convenience. The estimated total cost is $22000.

Table 2. Selected entries from full bill of materials

Component	Number	Cost per unit - US$	Total cost - US$	Source of materials	Material type
Dell Small Form Factor PC	1	1039.00	1039.00	Dell.com	Desktop PC
Intel NUC 10	1	326.02	326.02	Amazon.com	Portable PC
Ettus USRP B205 mini-i SDR	2	1571.00	3142.00	Ettus.com	SDR
ZHL-15W-422-S+	1	2639.25	2639.25	Minicircuits.com	High Power Amplifier
ZVE-8G+	1	1524.26	1524.26	Minicircuits.com	Power Amplifier
ZX60-83LN12+	2	211.19	422.38	Minicircuits.com	Low-noise Amplifier
DJI Matrice 600 Pro	1	4599.00	4599.00	Store.dji.com	Drone
RSA306B	1	5550.00	5550.00	Tek.com	Spectrum Analyzer
VBFZ-3590+	2	57.36	114.72	Mouser.com	Filter
VLF-4400+	2	33.46	66.92	Mouser.com	Filter
RM-WB1-DN-B1K	2	121.80	243.60	Mobilemarkantennas.com	Antennas

5. Build instructions

This section discusses the process of assembling and connecting hardware and installation and preparation of necessary software. Both PCs should have Ubuntu 18.04 installed before proceeding. Newer Ubuntu versions are also supported, but this was chosen when we started designing AERPAW in 2019. The experiment environment in this case lacks internet connectivity and requires transportation of all components to the location. Therefore, some steps are performed prior to the experiment and transportation while others are performed on-site immediately before the experiment begins. All steps that are required are discussed and can be performed as needed prior to the experiment.

5.1. Hardware

5.1.1 Cellular Ground Node

The Dell small form factor desktop PC is used as the companion computer for the fixed network node, executing the base station or the core network processes implemented in software. It comes prebuilt with all necessary components installed except the wireless network card. In order to connect the PC to the router, a wired ethernet connection can be used. To increase the overall range of the experiment, a WiFi card can be installed on the desktop PC. It is recommended to use a WiFi 6 capable card. This should be inserted into an available M.2 Key-E slot on the PC's motherboard when turned off. Once installed, connect the appropriate wires and antennas to the card. When the PC is turned on, the system should be able to search and connect to available WiFi networks. Additionally, one of the B205mini-i SDRs is plugged into any USB 3.0 port on the PC. If the SDR is plugged into a port that does not support USB 3.0, the device will still be detected by the system, but will fail to perform the experiment properly without USB 3.0 speeds.

The PA requires an external power supply. Figure 2 shows the power supply used for this experiment, LRS-350-24. The PA includes steps to attach RF cables and power in a specific order. These steps must be followed to avoid damage to any components. For the PA used in this experiment, the required steps are to attach RF input, apply power from the power supply, and attach RF output. In all cases, the filter in the transmit chain is placed closest to the RF input followed by the appropriate amplifier. On the TX chain, the RF input is considered the transmit port of the SDR. On the RX chain, the RF input is considered to be the antenna. The antennas used for the ground node are mounted on a speaker stand as shown in Figure 3 to allow for increased height off the ground and separation of antennas to prevent self-interference.

Fig. 2. 15 W power amplifier and power supply

Fig. 3. Speaker stand used to support antennas on the ground node with antenna separation to prevent self-interference.

5.1.2 Aerial Node Payload

The Intel NUC on the aerial node payload does not require any extra installation to take place except for RAM and hard drive that needs to be included. We use 32 GB RAM and a 500 GB hard drive which is an NVMe card installed in the M.2 slot. Smaller hard drive and 16 GB of RAM suffices for the proposed experiment. A larger hard drive is recommended if the aerial node is to be used for I/Q data collection. A perforated aluminum sheet is molded into an open-sided enclosure that will be held on the UAV. Carbon fiber poles are attached to the bottom corners of the aluminum. Each pole is about 2 feet long and is used to mount the antennas on its end. By mounting small and lightweight antennas at the ends of the poles, self-interference is reduced without adding too much weight to the payload. The NUC comes with a mounting kit that can be used to mount the device to the perforated sheet. If possible, the device should be mounted such that the ports are easily accessible for future use. The computer will need to be accessed for installation later. Alternatively, the installation process can be performed prior to mounting, but the device may still need to be accessed for future modifications. The LNA, PA, filters, and SDR should also be mounted to the sheet as appropriate using fasteners, zip ties, or other tools that will hold the components safely in place during flight. Similar to the desktop PC, the SDR must be connected to a USB 3.0 port to work as intended for this experiment. SMA cables of appropriate flexibility and lengths are used to connect each of the components on board. Figure 4 shows the overall implementation of the SDR system as the UAV payload to be easily mounted underneath a UAV. Figure 5 shows the top layer that includes the SDR, filters, PA, and LNA. The middle layer shown in Figure 6 consists of two power supplies, one for the NUC and one for the PA, a USB 3.0 splitter to power the SDR and other potential USB devices, and the mounts for the carbon fiber poles. The bottom layer does not include a perforated sheet and only consists of the NUC mounted upside-down to the bottom side of the middle layer's perforated sheet. Similar to the ground node, the filter is placed closest to the RF input followed by the appropriate amplifier. Due to the limited flight time of the UAV, in addition to further reductions caused by the weight of the payload, a separate battery is used for the payload from the UAV batteries. The 5s 3000 mAh battery powers the payload for multiple flights. A power converter is used to support the LNA and PA as both require the same voltage. An additional plug is available for transitioning between the battery and shore power. Once all devices are mounted and connected, the payload can be mounted onto the UAV using appropriate fasteners.

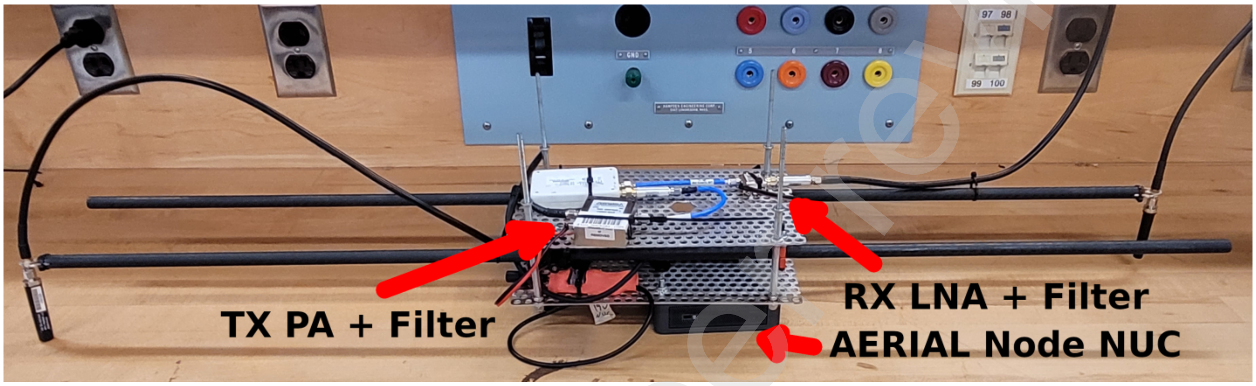

Fig. 4. Aerial node payload showcasing the RF front end, SDR and NUC computer

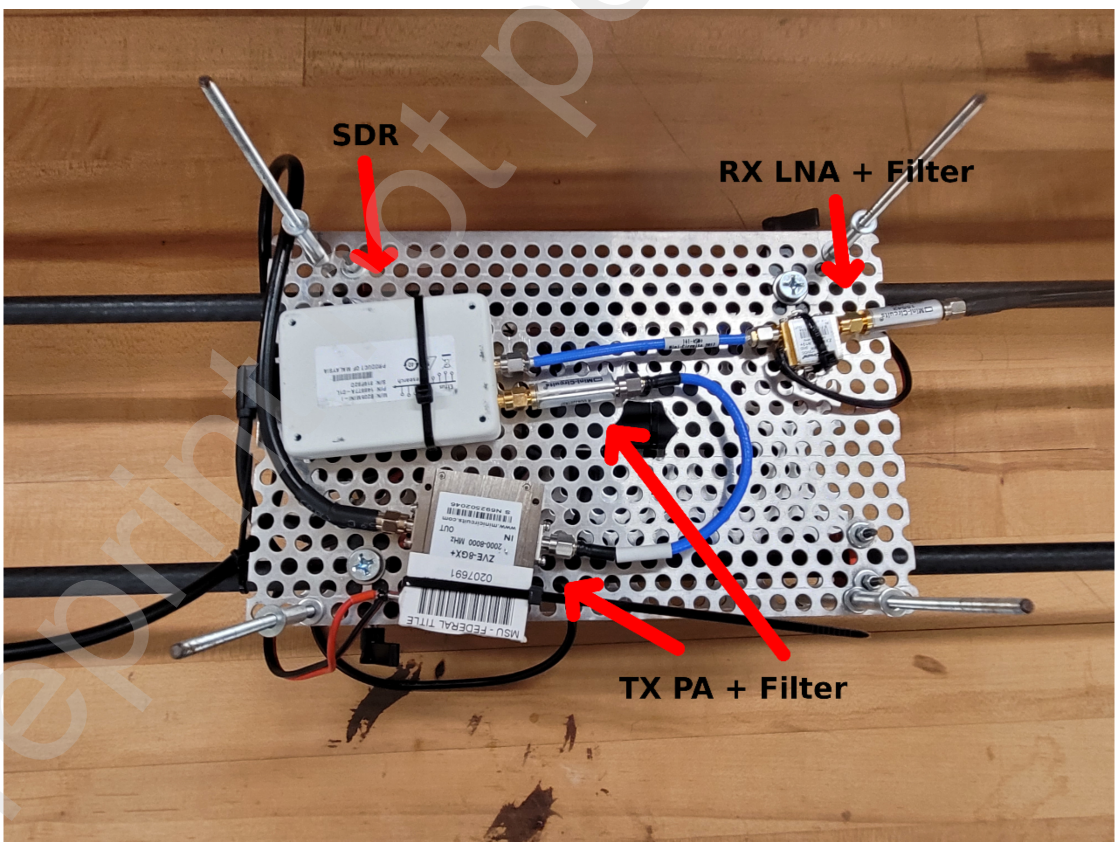

Fig. 5. Top view of the aerial node showcasing the top layer's SDR and RF front end.

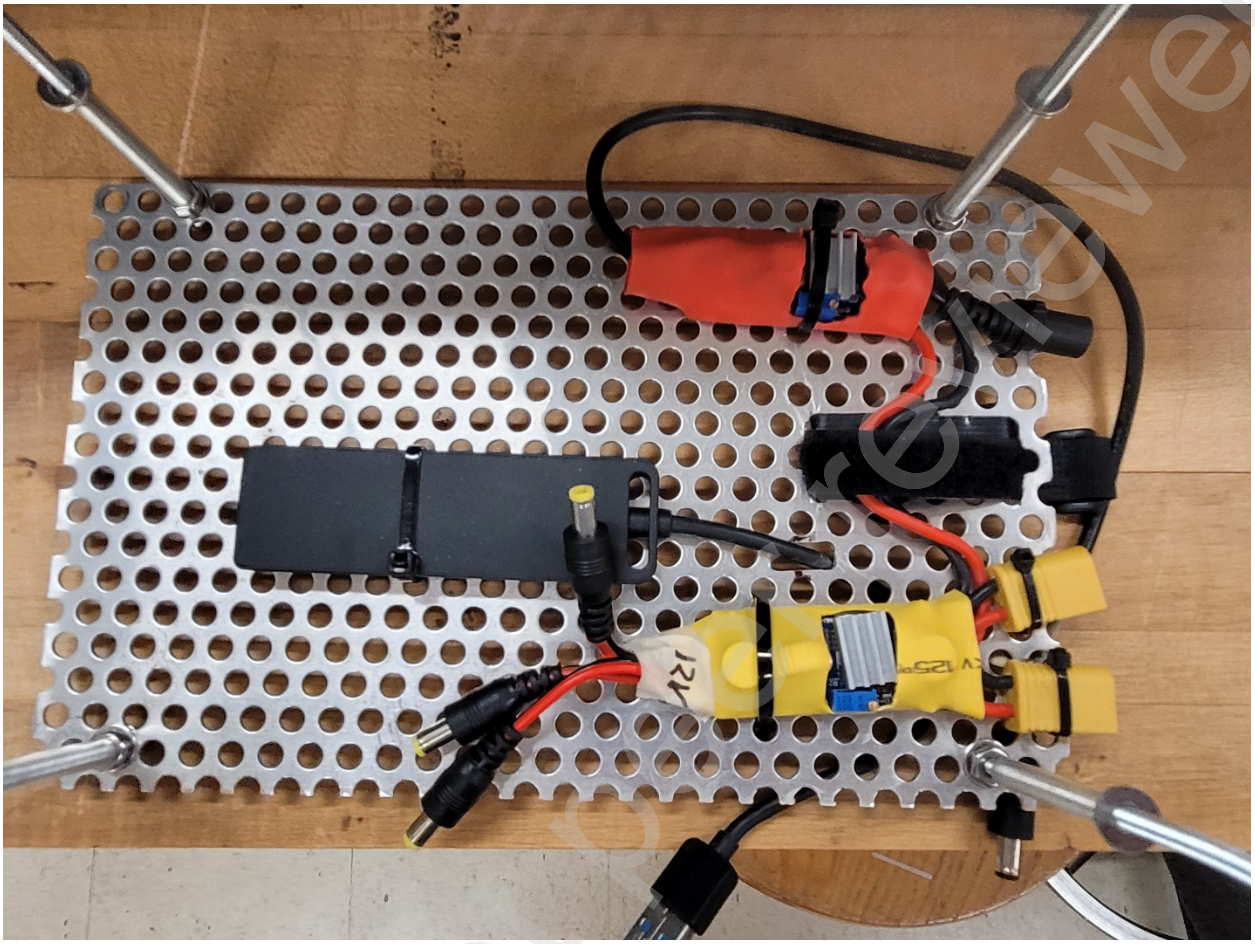
Fig. 6. Top view of the aerial node showcasing the middle layer's two power supplies and USB 3.0 splitter.

5.1.3 Spectrum Analyzer Node

While many types of spectrum analyzers are possible, the spectrum analyzer node that we use consists of the spectrum analyzer, a Windows laptop to run the spectrum analyzer software, Tektronix SignalVU, and the RFSpace TSA600-6000 antenna. The spectrum analyzer accepts a micro-USB 3.0 B cable and N-type male RF connector. An adapter is needed to connect the spectrum analyzer's N-type to the antenna's SMA connector. Finally, the spectrum analyzer must be connected to any of the laptop's USB 3.0 ports. This is required to receive the full 40 MHz of capture bandwidth.

5.1.4 WiFi Node

The WiFi node only consists of an Anker PowerHouse 200 battery pack and the WiFi router. No extra hardware installation is needed beyond plugging the WiFi router into the battery pack once the experiment is to be run.

5.2 Software

5.2.1 Desktop PC

To begin the software installation process, the desktop PC should have Ubuntu 18.04 installed and a stable internet connection available. If a WiFi network is available, a connection can be established to verify the functionality of the newly installed WiFi card. The first software solution that must be installed is UHD. This driver allows for the PC and other appropriate software to communicate with the available SDRs. The UHD install process is outlined in the UHD and USRP manual [6]. A simplified process to install and use the version selected for this experiment is shown in Figure 7. Once UHD is installed, both srsRAN and OAI can be installed. The installation process for srsRAN is outlined on their documentation website [7]. Figure 8 shows the commands required to install srsRAN. The install process for OAI is discussed on their repository [8]. This experiment only requires the eNB to be installed from OAI, as shown in Figure 9. Other useful tools that should be installed include iperf3, openssh-server, any command line text editor, such as vim. Iperf3 allows for throughput measurements to be performed over the SDR network. Openssh-server allows for each PC to use the ssh tool to connect to each other. Vim is a text editor that can be used to modify configuration files with a graphical interface. These tools can be installed using: sudo apt-get install iperf3 openssh-server vim. Both srsRAN and OAI require modified configurations from their default installation. The files needed are located with the available design files. The desktop PC will need all files from the srsRAN folder and the enb.band7.tm1.100PRB.usrpb210.conf file in the OAI folder. The files used for srsRAN can be copied to the following path: ~/.config/srsran/. The configuration file from the OAI folder may be copied anywhere on the device.

```
#Install dependencies
sudo apt-get install autoconf automake build-essential ccache cmake cpufrequtils doxygen \
ethtool g++ git inetutils-tools libboost-all-dev libncurses5 libncurses5-dev libusb-1.0-0 \
libusb-1.0-0-dev libusb-dev python3-dev python3-mako python3-numpy python3-requests \
python3-scipy python3-setuptools python3-ruamel.yaml

#Clone and open UHD repository and select version
git clone https://github.com/EttusResearch/uhd.git
cd uhd
git checkout v4.0.0.0

#Set up build directory
cd host
mkdir build
cd build

#Check dependencies and build software for install
cmake ../
make -j$(nproc)
sudo make install

#Setup library path for commands to be run
sudo ldconfig

#Acquire FPGA images for SDRs
uhd_images_downloader

#Check for connected SDRs (This will return information about any USRP SDRs connected)
uhd_find_devices
```

Fig. 7. UHD install process. This will install the UHD version that is used for the experiment outlined

```
#Install dependencies
sudo apt-get install build-essential cmake libfftw3-dev libmbedtls-dev \
libboost-program-options-dev libconfig++-dev libsctp-dev

#Clone srsRAN 4G repository and select version
git clone https://github.com/srsRAN/srsRAN_4G.git
cd srsRAN_4G
git checkout v20.10.1

#Set up build directory
mkdir build
cd build

#Check dependencies and builds software for install
cmake ../
make -j$(nproc)
sudo make install

#Setup library path for commands to be run and install default configuration files
sudo ldconfig
srsran_4g_install_configs.sh user
```
Fig. 8. srsRAN install process. This will install the srsRAN version that is used for the experiment outlined

```
#Clone and open OAI repository
git clone https://gitlab.eurecom.fr/oai/openairinterface5g.git
cd openairinterface5g

#Select software version
git checkout v1.2.2

#Set up build directory
mkdir build
cd build

#Install dependencies and eNodeB
./build_oai -I --eNB

#Rebuilds eNodeB with using UHD for use with USRP SDRs
./build_oai -w USRP –-eNB
```
Fig. 9. OAI install process. This will install the OAI version that is used for the experiment outlined

5.2.2 Aerial Node PC

The aerial node PC will follow the same software installation processes as the desktop PC with the exception of OAI. OAI is only used for the eNB and is unnecessary for the aerial node PC as it will act as a UE for this experiment. It is also recommended that this PC is connected to the WiFi network that will be used for the experiment prior to being flown. This will ensure that it attempts to reconnect to the same network if it is rebooted. When the UAV is flying with the aerial node payload, the PC will not have any available peripherals and will only be accessible using the ssh tool from the desktop PC on the same WiFi network. The file needed are located with available design files. This PC only needs the ue.conf to be downloaded and located on the NUC for the here described experiment. The srsRAN UE will be used with both the srsRAN and OAI eNBs.

5.2.3 Spectrum Analyzer Node

The laptop at the spectrum analyzer node must use Microsoft Windows in order to use Tektronix SignalVU. Two displays are used to record data from the spectrum analyzer, default spectrum and channel power and adjacent channel power ratio (ACPR). The default spectrum simply shows the signal that is being received over the selected bandwidth. The frequency is set for 3.51 GHz for viewing downlink and 3.41 GHz for viewing uplink. Using the settings cog wheel in the bottom-right of the program window, other options can be selected such as max hold or various averaging options. By default, a capture bandwidth of 40 MHz is used to see the full spectrum beyond the experiment's 20 MHz bandwidth transmission. The channel power and ACPR mode allows for the average and peak channel power to be measured. This can be selected using the "Setup" option on the toolbar, selecting "Displays", then "RF Measurements", then selecting the display or interest, "Chan Pwr and ACPR". This display can be added as an additional display side-by-side the spectrum display Once selected, the settings cog wheel can be used to modify options of this mode. The options of interest are the bandwidth and adjacent channels. In this case, the actual experiment bandwidth is chosen, 20 MHz for the default experiment, with zero adjacent channels. Figure 10 shows example output utilizing both the spectrum and channel power and ACPR displays.

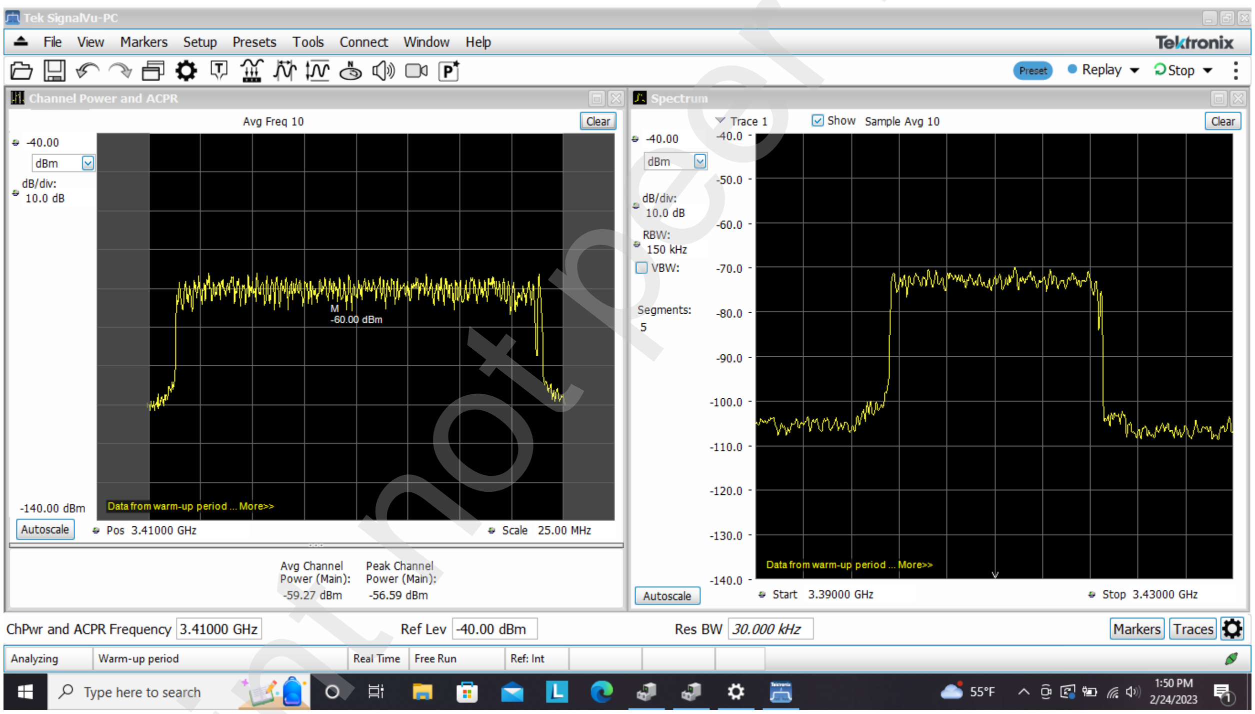

Fig. 10. Example output from SignalVU featuring the default spectrum view (right) and the channel power and ACPR view (left).

5.2.4 WiFi Node

To reduce potential interference caused by the router, the 2.4 GHz band will be disabled and the 5 GHz band will be restricted to a single channel. To access the WiFi router, a PC with a connection to router's network, wired or wireless, must be navigated to the router's IP address, "192.168.0.1", using any browser. The wireless settings are accessed via "Settings" followed by "Wireless". Both WiFi Mesh and Smart Connect should be disabled. This will reveal specific band options for 2.4 GHz and 5 GHz. From here, the 2.4 GHz band can be disabled as well. Under 5 GHz, advanced settings can be selected to reveal several more options. The "Wi-Fi Channel" option will be set to "Auto" by default. This is changed to the highest value available, channel 165, to create the largest distance from the experimental frequency. Once complete, make sure to save the settings by clicking "Save" at the top of the page. The router page will become unavailable while the changes are made. Figure 11 shows the settings used for the D-Link X5460 router.

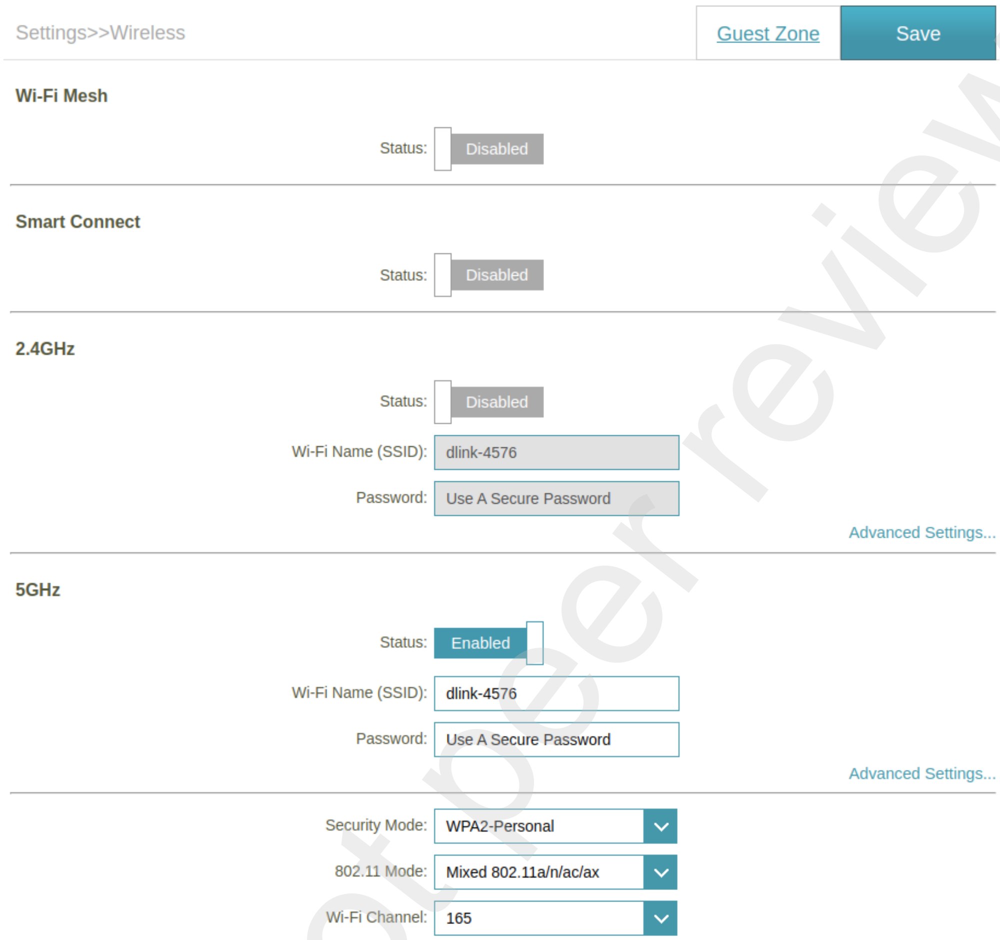

Fig. 11. Settings used for the D-Link X5460 WiFi Router

6. Operating instructions

Once all of the hardware and software is properly assembled and installed, the initial experiment steps can begin. The required software files located with the available design files should have been downloaded onto their required systems at this point. The first step is to determine the IP address of the NUC on the aerial node. This can be done by logging into the WiFi router and investigating the connected users. Figure 12 shows an example output from the WiFi router's connected users page. Based on the example figure, open a new terminal and use the following command to connect to the nuc10 device shown.

```
ssh nuc10@192.168.0.113
```

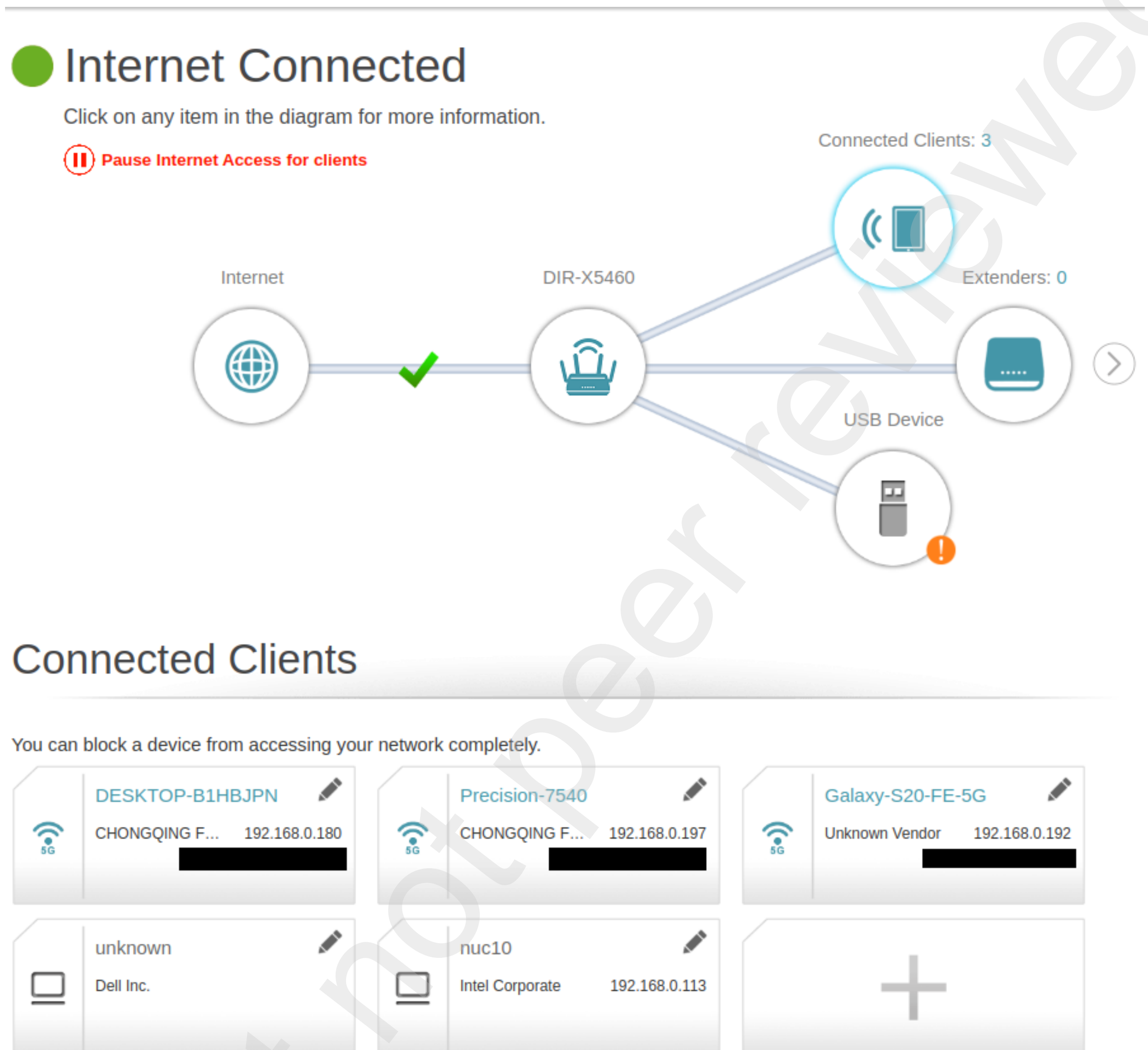

Fig. 12. Example output of connected users from the WiFi router

If successful, the username displayed by the terminal should be the username of the computer being connected to, "nuc10" for the example command. This terminal that is now controlling the NUC will be considered the aerial node PC or NUC. On both the ground node PC and aerial node PC, the following command should be run to change the CPU governor to performance mode. The exact same commands should be executed on the desktop PC used for the fixed SDR node. This instructs the CPU cores to run at their highest clock speed, improving the performance with the SDRs.

```
for ((i=0;i<$(nproc --all);i++)); do sudo cpufreq-set -c $i -r -g performance; done
```

At this point, both PCs are fully prepared to begin the experiment. To start the core network, the following command should be run in a new terminal on the ground node PC. Figure 13 shows the expected output of this command.

```
sudo srsepc
```

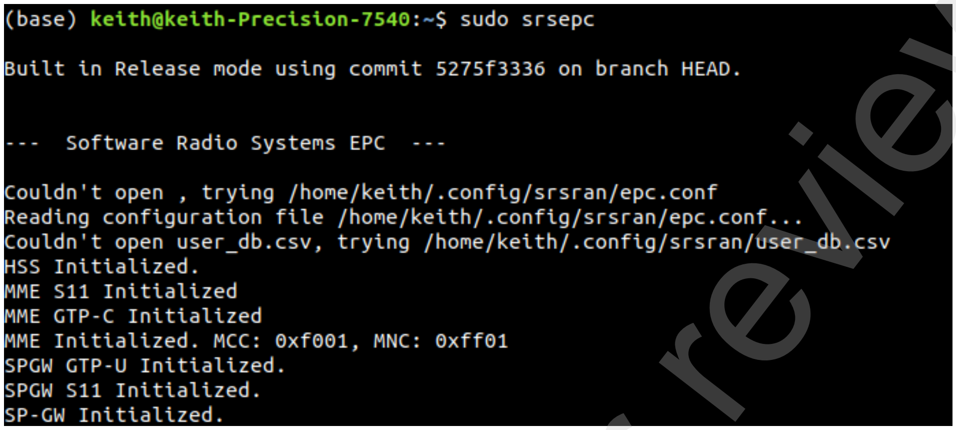

Fig. 13. Output from starting the core network showing the initialization of the essential LTE core network functions.

In a separate terminal on the ground node PC, run the following command to start srsENB. Figure 14 shows the expected output that will occur from the terminal running srsEPC. This indicates that the base station successfully connected to the EPC. Figure 15 shows the expected output from the terminal starting srsENB. The output shows firmware and FPGA images being loaded to the SDR. These messages will only occur the first time an SDR is run after being powered on.

```
sudo srsenb
```

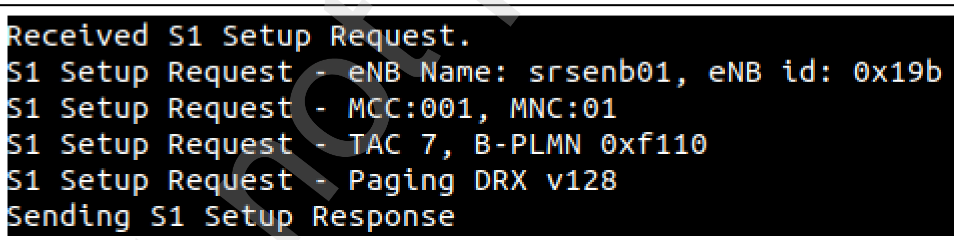

Fig. 14. Output indicating that srsENB has successfully connected to srsEPC.

Alternatively, the OAI base station can be used instead by running the following commands.

```
cd ~/openairinterface5g/cmake_targets/lte_build_oai/build
sudo -E ./lte-softmodem -O <location of enb.conf from design files>
```

In the terminal that is connected to the aerial node PC, run the following command to start srsUE. Figure 16 shows the expected output of this command before a connection is established.

```
sudo srsue
```

```
(base) keith@keith-Precision-7540:~$ sudo srsenb
---  Software Radio Systems LTE eNodeB  ---

Couldn't open , trying /home/keith/.config/srsran/enb.conf
Reading configuration file /home/keith/.config/srsran/enb.conf...
Couldn't open sib.conf, trying /home/keith/.config/srsran/sib.conf
Couldn't open rr.conf, trying /home/keith/.config/srsran/rr.conf
Couldn't open rb.conf, trying /home/keith/.config/srsran/rb.conf

Built in Release mode using commit 5275f3336 on branch HEAD.

/home/keith/workarea/srsRAN/srsenb/src/enb_cfg_parser.cc:1216: Force DL EARFCN for cell PCI=1 to 6700
Opening 1 channels in RF device=default with args=default
Available RF device list: UHD  zmq
Trying to open RF device 'UHD'
[INFO] [UHD] linux; GNU C++ version 7.5.0; Boost_106501; UHD_4.2.0.HEAD-0-g46a70d85
[INFO] [LOGGING] Fastpath logging disabled at runtime.
[INFO] [B200] Loading firmware image: /usr/local/share/uhd/images/usrp_b200_fw.hex...
[INFO] [B200] Loading firmware image: /usr/local/share/uhd/images/usrp_b200_fw.hex...
Opening USRP channels=1, args: type=b200,master_clock_rate=23.04e6
[INFO] [UHD RF] RF UHD Generic instance constructed
[INFO] [B200] Detected Device: B205mini
[INFO] [B200] Loading FPGA image: /usr/local/share/uhd/images/usrp_b205mini_fpga.bin...
[INFO] [B200] Operating over USB 3.
[INFO] [B200] Initialize CODEC control...
[INFO] [B200] Initialize Radio control...
[INFO] [B200] Performing register loopback test...
[INFO] [B200] Register loopback test passed
[INFO] [B200] Asking for clock rate 23.040000 MHz...
[INFO] [B200] Actually got clock rate 23.040000 MHz.
RF device 'UHD' successfully opened

==== eNodeB started ===
Type <t> to view trace
Setting frequency: DL=3510.0 Mhz, UL=3410.0 MHz for cc_idx=0 nof_prb=100
```

Fig. 15. Output from starting the base station for the first time since SDR power on

```
(base) keith@keith-Precision-7540:~$ sudo srsue
Couldn't open , trying /home/keith/.config/srsran/ue.conf
Reading configuration file /home/keith/.config/srsran/ue.conf...

Built in Release mode using commit 5275f3336 on branch HEAD.

Opening 1 channels in RF device=default with args=default
Available RF device list: UHD  zmq
Trying to open RF device 'UHD'
[INFO] [UHD] linux; GNU C++ version 7.5.0; Boost_106501; UHD_4.2.0.HEAD-0-g46a70d85
[INFO] [LOGGING] Fastpath logging disabled at runtime.
Opening USRP channels=1, args: type=b200,master_clock_rate=23.04e6
[INFO] [UHD RF] RF UHD Generic instance constructed
[INFO] [B200] Detected Device: B205mini
[INFO] [B200] Loading FPGA image: /usr/local/share/uhd/images/usrp_b205mini_fpga.bin...
[INFO] [B200] Operating over USB 3.
[INFO] [B200] Initialize CODEC control...
[INFO] [B200] Initialize Radio control...
[INFO] [B200] Performing register loopback test...
[INFO] [B200] Register loopback test passed
[INFO] [B200] Asking for clock rate 23.040000 MHz...
[INFO] [B200] Actually got clock rate 23.040000 MHz.
RF device 'UHD' successfully opened
Waiting PHY to initialize ... done!
Attaching UE...
```

Fig. 16. Output from starting the user equipment for the first time since SDR power on

If the core network, base station, and UE processes are all started as expected, the aerial node PC terminal will indicate that it successfully connected to the network and has been assigned an IP address in the experimental LTE network. Figure 17 shows example output from this connection. Both the core network and base station should indicate that a new connection has been established as well.

```
Attaching UE...
.
Found Cell:  Mode=FDD, PCI=1, PRB=100, Ports=1, CP=Normal, CFO=0.2 KHz
Found PLMN:  Id=00101, TAC=7
Random Access Transmission: seq=14, tti=3221, ra-rnti=0x2
RRC Connected
Random Access Complete.     c-rnti=0x46, ta=1
Network attach successful. IP: 172.16.0.2
Software Radio Systems RAN (srsRAN) 23/3/2023 0:38:59 TZ:0
```

Fig. 17. Expected output from a successful connection between the UE and the network

Now that the full end-to-end system is working as expected, data about the link can be gathered and traffic can be passed over the network. To gather trace data about the connection between the base station and UE, click the terminal that is running srsENB, type ‘t’ followed by the Enter key. The base station terminal will begin to report information about the connection. An example of the trace reported by srsENB is shown in Figure 18. This process can be repeated on srsUE to report information from its end, shown in Figure 19. Some different network measurements are performed on the UE such as reference signal received power (RSRP) and signal-to-noise ratio (SNR).

```
          ----------------DL---------------|-------------------------UL-----------------------
rat rnti  cqi  ri  mcs  brate   ok  nok  (%) | pusch  pucch  phr  mcs  brate   ok  nok  (%)    bsr
lte   47   15   0    0    776    1    0   0% |  32.8    6.7   40   12   4.4k    3    0   0%    0.0
lte   47   15   0    0    776    1    0   0% |   n/a    5.9   40    0   4.4k    3    0   0%    0.0
lte   47   15   0    0    776    1    0   0% |  33.2    7.0   40   14   4.4k    3    0   0%    0.0
lte   47   15   0    0    776    1    0   0% |  32.5    6.0   40   12   4.4k    3    0   0%    0.0
lte   47   15   0   12    776    1    0   0% |  33.1    5.8   40   12   4.4k    3    0   0%    0.0
lte   47   15   0    0    776    1    0   0% |   n/a    7.1   40    0   4.4k    3    0   0%    0.0
lte   47   15   0    0    776    1    0   0% |  33.0    7.3   40   14   4.4k    3    0   0%    0.0
lte   47   15   0    0    776    1    0   0% |  32.8    3.6   40   12   4.4k    3    0   0%    0.0
```

Fig. 18. Trace reported by srsENB

```
---------Signal----------|----------------DL----------------|-----------UL----------
 cc  pci  rsrp   pl   cfo | mcs  snr  iter  brate  bler  ta_us | mcs  buff  brate  bler
  0    1   -70   70   358 |  12   30   1.0    776    0%    0.5 |  15   0.0   4.4k    0%
  0    1   -70   70   360 |  12   30   1.0    776    0%    0.5 |  15   0.0   4.4k    0%
  0    1   -70   70   372 |  12   30   1.0    776    0%    0.5 |  15   0.0   4.4k    0%
  0    1   -70   70   388 |  12   30   1.0    776    0%    0.5 |  15   0.0   4.4k    0%
  0    1   -69   69   389 |  12   31   1.0    776    0%    0.5 |  15   0.0   4.4k    0%
  0    1   -70   70   389 |  12   30   1.0    776    0%    0.5 |  15   0.0   4.4k    0%
```

Fig. 19. Trace reported by srsUE

By default, the UE will disconnect from active mode and go into idle mode after a short period of time if there is no traffic being sent over the network. This is done to conserve power when there is not a need to maintain an active data session with the network. The connection can be reestablished by sending traffic over the network. Note that if the connection is reestablished without restarting srsEPC, a new IP address will be assigned to srsUE. Two options are presented for creating traffic over the network to test its latency and throughput capacity, ping and iperf3. Ping is used to create minimal traffic over the network to maintain a connection and determine latency. The following command can be used to run ping over the network in a new terminal.

```
ping <IP address>
```

This command can be run from the ground node PC by using the UE's IP address, 172.16.0.2 as shown in Figure 17. Alternatively, ping can be performed starting from the UE by using ssh to access the aerial node PC via a second terminal and using the above command with the core network's IP address, 172.16.0.1. To test the throughput capacity of the network, iperf3 is used. To run iperf3, a server and client node must be established using the following commands.

Server:

```
iperf3 -s
```

Client:

```
iperf3 -c <iperf3 server IP address>
```

By default, the traffic from iperf3 will flow from the client to the server. Therefore, to test the throughput capacity of the downlink connection, the iperf3 server should be run on the ground node PC and the client should be run on the aerial node PC. The iperf3 client and server can be switched to test the uplink capacity of the connection by appending "-R" to the client command. This instructs iperf3 to send traffic in the reverse direction from server to client. The iperf3 server IP address will match the IP address used for ping, 172.16.0.1 for the core network and 172.16.0.2 for the UE.

7. Validation and characterization

The system can be tested by passing traffic into the system to be communicated over the newly created network and by flying the UAV to determine how height and range may impact the results. Validation is performed by flying the UAV to various heights and ranges and holding the vehicle in place. While held in place, iperf3 is used to test the throughput capacity between the ground node and aerial node. The system that is described for this experiment is also used by Abdalla et al. [9]. Abdalla et al. shows results following this process for varying distances and heights using iperf3 traffic described previously. Figure 20 shows the experimental setup being run with the ground node PC and RF front end while the aerial node flies in the distance.

In addition to the default experiment described in this article, many experimental parameters can be modified to extend existing use cases. One potential modification to this experiment may include varying individual SDR transmit and receive gains to monitor their impact to the channel capacity and potential distortion cause to surrounding signals. This can be done by modifying the srsENB configuration file parameters "tx_gain" and "rx_gain". To use these gains properly, it is important to know the range for the SDR being used, 0-89.8 and 0-76 for the transmit and receive gains of the B205mini-i, respectively. The bandwidth of the signal can also be changed by modifying "n_prb" in the srsENB configuration file to any value supported by srsRAN.

Example results using a 10 MHz bandwidth (n_prb = 50) are shown in Figure 21. For these results, the UAV was consistently flown at a height of 25 m. While the UAV is close to the base station, the throughput values are not far from the expected maximum values of 35 Mbps and 25 Mbps for downlink and uplink, respectively. As the horizontal distance from the base station to the UE increases, the throughput decreases because of the increasing path loss and decreasing SNR at the receiver. For each measurement point, iperf3 is used to measure the throughput capacity in a single direction, uplink or downlink. It is also possible to run uplink and downlink iperf3 simultaneously due to the experiment being setup for frequency division duplex (FDD). Therefore, the uplink and downlink signals occur on separate frequencies allowing for constant traffic flow in both directions.

More complex modifications to the experiment can be done such as having multiple users and base stations, handover, and other transmission modes, such as 2x2 MIMO. These require changes to several files, new software installations, and changes to the aerial and ground node hardware setups. The experimental setup in this article gives a baseline setup required to perform an end-to-end experiment using SDR hardware and open-source cellular network software and can be directly extended to accommodate other experiments or use cases.

Fig. 20. Outdoor experimental setup in use showing ground node and aerial node

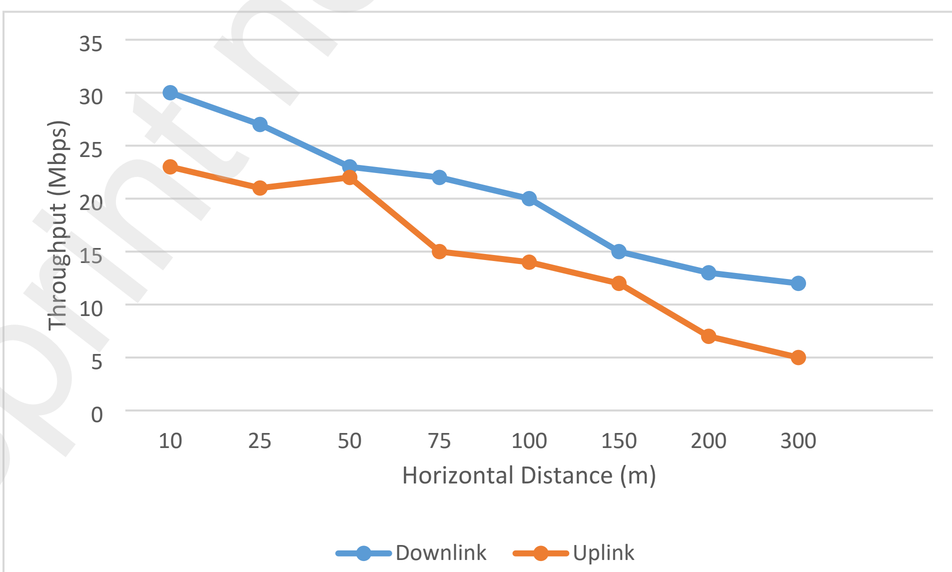

Fig. 21. Throughput results using experimental setup with 10 MHz bandwidth and UAV at 25m height

Ethics statements

Declaration of Competing Interest
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Human and Animal Rights
Not applicable.

Credit author statement
Keith Powell: Conceptualization, Methodology, Software, Validation, Formal Analysis, Investigation, Visualization, Writing-Original Draft. **Andrew Yingst:** Conceptualization, Validation, Investigation, Writing-Review and Editing. **Vuk Marojevic:** Supervision, Project Administration, Funding Acquisition, Writing-Review and Editing.

Acknowledgments
K. Powell, A. Yingst, and V. Marojevic are supported in part by the National Science Foundation under grant number CNS-1939334.

References:

[1] "LTE fixed wireless equipment," *BLiNQ Networks*, 29-Nov-2022. [Online]. Available: https://blinqnetworks.com/products/. [Accessed: 17-Feb-2023].

[2] "Products," *PRODUCTS – Airspan.com*. [Online]. Available: https://www.airspan.com/products/. [Accessed: 18-Feb-2023].

[3] *Ukama*. [Online]. Available: https://www.ukama.com/. [Accessed: 18-Feb-2023].

[4] EttusResearch, "EttusResearch/UHD: The USRP™ hardware driver repository," *GitHub*. [Online]. Available: https://github.com/EttusResearch/uhd. [Accessed: 28-Feb-2023].

[5] Srsran, "Srsran/srsran_4G: Open source SDR 4G software suite from Software Radio Systems (SRS)," *GitHub*. [Online]. Available: https://github.com/srsran/srsRAN_4G. [Accessed: 03-Mar-2023].

[6] "Building and Installing UHD from source," *USRP Hardware Driver and USRP Manual*. [Online]. Available: https://files.ettus.com/manual/page_build_guide.html. [Accessed: 05-Mar-2023].

[7] "Installation guide," *Installation Guide - srsRAN 4G 22.10 documentation*. [Online]. Available: https://docs.srsran.com/projects/4g/en/latest/general/source/1_installation.html. [Accessed: 05-Mar-2023].

[8] "Doc/BUILD.md · develop · OAI / openairinterface5G · GITLAB," *GitLab*. [Online]. Available: https://gitlab.eurecom.fr/oai/openairinterface5g/blob/develop/doc/BUILD.md. [Accessed: 09-Mar-2023].

[9] A. S. Abdalla, A. Yingst, K. Powell, A. Gelonch-Bosch and V. Marojevic, "Open Source Software Radio Platform for Research on Cellular Networked UAVs: It Works!," in IEEE Communications Magazine, vol. 60, no. 2, pp. 60-66, February 2022, doi: 10.1109/MCOM.001.2100394.