MITRE ENGENUITY | Open Generation

EXPERIMENT REPORT

EXPLORE 5G CONNECTIVITY FOR UAS COMMAND AND CONTROL

by Dipesh Modi (MITRE), Rick Niles (MITRE), Izabela Gheorghisor (MITRE), Angela Chen (MITRE), Aloizio Pereria da Silva (Virginia Tech), Ying Wang (Virginia Tech), Shereef Sayed (MITRE), and Leila Z Ribeiro (MITRE)

Contents

Introduction

MITRE Engenuity Open Generation brings together diverse technical viewpoints and domain expertise from across industries and market verticals, to collaboratively design, develop, and demonstrate connected-world solutions that may be uniquely enabled by 5G capabilities. The mission is to create public good, advance the commercialization of 5G enterprise use cases, and unlock massive economic value in the U.S.

As part of Open Generation's efforts to accelerate life-changing 5G applications, MITRE Engenuity and Virginia Tech are collaboratively working on a series of testbed experiments designed to enable and advance the performance of 5G technology utilization for Uncrewed Aerial Systems (UAS)—also known as drones.

This document describes the experiments executed for Phase 1B of the testbed experiment series, as agreed between MITRE Engenuity (ME) and Virginia Tech (VT).

Background: Exploring 5G Connectivity for UAS Command and Control

The primary objective of the ongoing ME/VT experimentation series is to enable drone use cases that require the ability to safely operate beyond visual line of sight (BVLOS). As such, the series of experiments focuses on command and control (C2) link connectivity, including assessing and improving the link's availability with respect to area, altitude, and mobility. Assessing payload (i.e. video streaming) performance is a related, secondary objective in the early phases.

The initial phases (1A, 1B) of the experiments executed to date focused on demonstrating proof-of-concept and assessing a baseline achievable performance of 5G for C2-link drone functions and for concurrent video stream traffic.

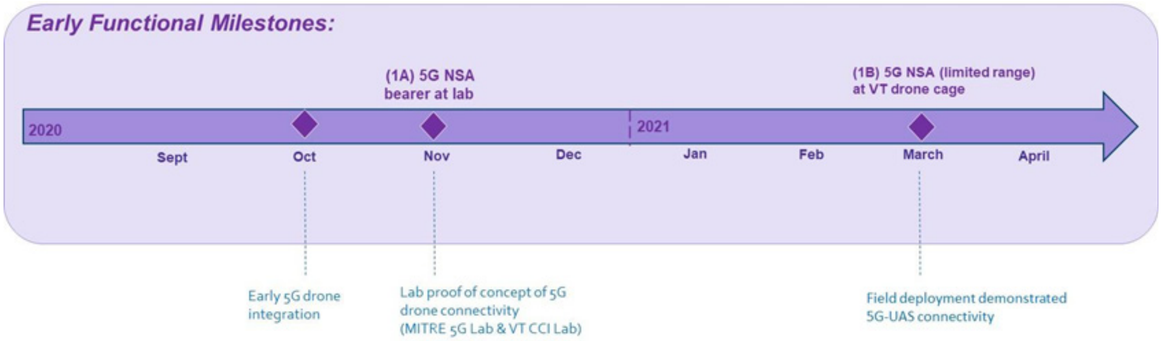


FIGURE 1. NOTIONAL TIMELINE OF COMPLETED OPEN GENERATION UAS C2 EXPERIMENTS AT VIRGINIA TECH, THROUGH PHASE 1B

Phase 1A was strictly a proof of concept on the drone design, its integration with 5G User Equipment (UE), and the completion of an end-to-end C2 link over 5G at the lab environment. Successfully completed in November 2020, Phase 1A utilized a setup allowing a Ground Control Station (GCS) to control, send commands to, and receive sensor data from a drone C2 link utilizing a 5G Non-Stand-Alone (NSA) Radio. The experiments were performed both at the MITRE 5G Testbed Lab, and at VT's Commonwealth Cyber Initiative (CCI/VT)'s 5G Testbed Lab.

Phase 1B, which is covered in the scope of this document, focused on performing the proof of concept in the field, specifically at VT's Mid-Atlantic Aviation Partnership (MAAP/VT) Drone Park facility in Blacksburg, VA. Drone testing was carried out by CCI/VT and MITRE personnel. This was a limited test using low-power, private equipment with limited range.

At the conclusion of this proof-of-concept phase, the team accomplished the following:

- Integrated drone with 5G user equipment (UE)
- Implemented end-to-end communications between ground control station (GCS) and drone connected to a 5G NSA base station, including:
  - Drone C2 telemetry (sensor data from drone to GCS)
  - Drone C2 commands (commands from GCS to drone)
  - Real-time, in-flight video streamed from drone, to ground server
- Performed flight tests with the 5G drone operating in the following modes:
  - Loiter (held position and altitude based on GPS & barometric altimeter)
  - Auto (flight along a predefined path)
  - Guided (sent commands to proceed to each waypoint in real time, over cellular connection)

Additional experimentation activities are being planned at VT and at other Open Generation testbed locations[1]. These future activities will focus on enabling specific drone use cases and improving performance by implementing drone-specific mechanisms and advanced 5G capabilities. Those activities are beyond the scope of this document[2].

Phase 1B Scope, Objectives, and Activities

The next sections describe Phase 1B test objectives and main activities to serve as a reference if such experiment is repeated, in full or partially. Detailed test procedures, logistics, and materials leading up to test day, data collected, post-processing activities, and sample results are also included.

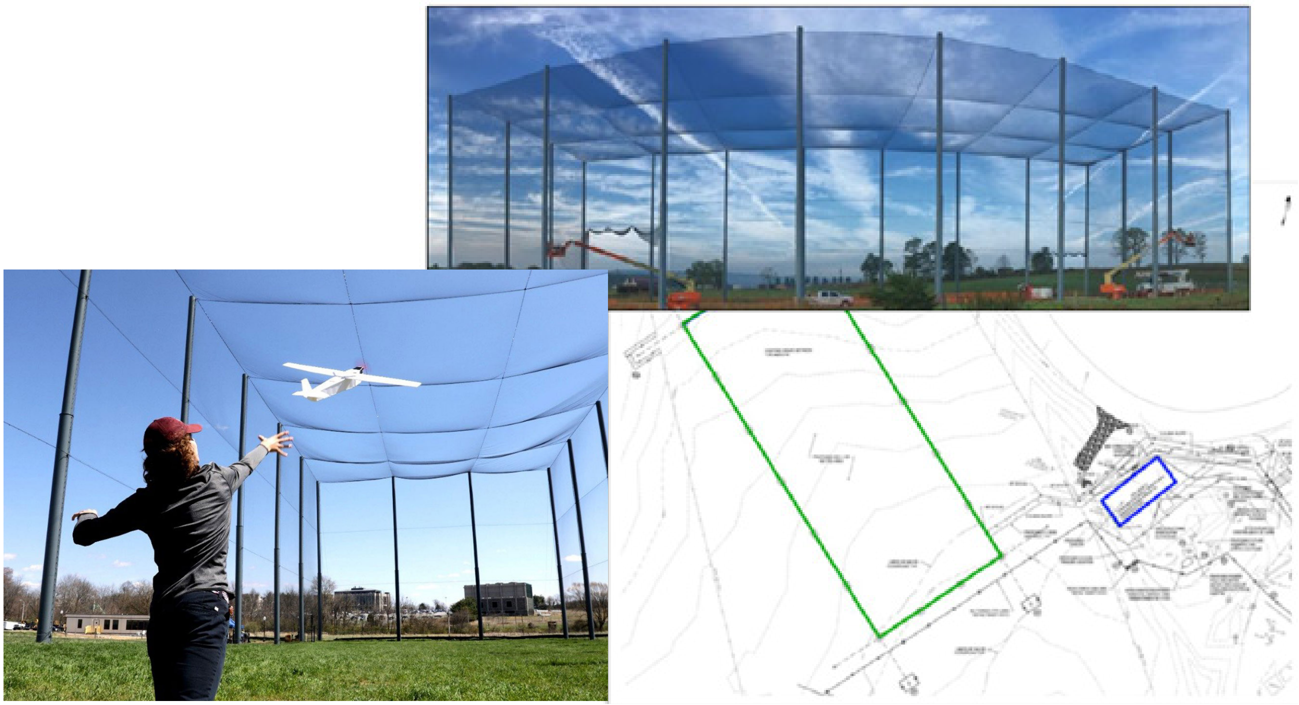


FIGURE 2. VT'S DRONE CAGE FACILITY (PHOTO BY MAAP, VT)

Specific highlights related to scope of the experiment:

- Dates and Location: Tests were performed from March 2 to 5, at VT's Drone Cage facility, located in Blacksburg, VA, adjacent to VT's main campus. That facility is illustrated in Figure 2.
- Given the enclosed nature of the drone cage (dimensions 300'L x 120'W x 80'H), tests were performed at low altitude and within short range from the base station.
- The Device and System Under Test (DUT/SUT) was limited to the Uncrewed Aircraft (UA)/drone and the underlying 5G network in NSA architecture.
- The 5G network deployment followed NSA Architecture Option 3x. LTE was used as the control plane anchor for NR. User data traffic flowed directly to the gNB part of the base station.
- The bands utilized for the experiment were B1 band (2.1 GHz) for LTE, and n78 band (3.5GHz) for 5G-NR.
- The 5G network implementation was limited to one portable unit, the Amari Callbox (5G Network in a box), which provided the radio access network (RAN) and Core network functions. This unit had limited output power and provided short coverage range (further detail in the 5G RAN and EPC (Amari Callbox) section.

One of the key goals for the Phase 1B experiment was to implement a proof-of-concept end-to-end 5G UAS solution, and to compare performance of drone connectivity using 5G NR and LTE waveforms. This effort included these key activities:

- Developed and documented the solution, configuration details, and input parameters.
- Measured link performance with 5G NR and LTE waveforms.
- Recorded statistics using the Amari Callbox and a MITRE-developed Android app (denoted as "Cell Info Logger" within this document) for mobile UE.
- Post-processed collected data for reporting.

Pre-tests were needed to check the ability to collect and save metrics. The following connectivity pre-tests were performed:

- On the downlink (connectivity from UA), captured synchronization signal (SS) reference signal received power (RSRP) and reference signal received quality (RSRQ) at drone.
- On the uplink (connectivity from UA), captured telemetry and video data.
- Performed multiple tests to collect sample data sets over multiple flights:
  - Flight at two different UA altitudes, following the same flight pattern over the same grounds, to test variability in propagation channel under the same conditions.
  - Flight at two different UA altitudes, following different flight patterns over different grounds (open range), to test variability in radio frequency (RF) parameters impact / antenna pattern.

The following assumptions were made:

- Fine tuning of 5G Quality of Service (QoS) or prioritizing payload categories—including C2, telemetry data, and real-time video streaming—was not in scope during Phase 1B of testing. Default QoS was provided by Amari Callbox for all traffic types.
- The aircraft's UE did not act as a server (i.e., we did not "host" services on the remote endpoints).

Test Strategy

Overview UAS and 5G

UAS MODEL IN 3GPP ECOSYSTEM

As previously mentioned, Phase 1B experimentation and reporting described in this document is focused on early prototyping and proof of concept efforts for future testing of drone connectivity using 5G.

Among the standardization activities in the Third Generation Partnership Project (3GPP), one area of focus has been the work on enhancements for 5G systems to support UAS connectivity needs. Specifically, 5G systems must safely support drones and UA controllers (denoted as ground control stations in this document). UAS operations are also envisioned to take place within a UAS Traffic Management (UTM) environment as described in the UTM Concept of Operations.[3]

As 5G use cases continue to evolve and new use cases are being developed, 3GPP continues to identify new features and enhancements needed—not just for the safe operation of drones, but also to ensure that other users of the network do not experience a degradation of service when drones are operating nearby.

Drone-related efforts started in 3GPP Release 15 with the development of the technical report on Enhanced LTE Support for Aerial Vehicles[4] followed by subsequent enhancements in technical specifications.[5] Work continued in Releases 16 and 17 with most recent requirements and enhancements being defined in technical specifications.[6,7]

A high-level overview of drone-related efforts within 3GPP is also available.[8] Drone-related activities in 3GPP continue and are mentioned in this report to provide context for the state of 5G systems and architecture at the time of experimentation.

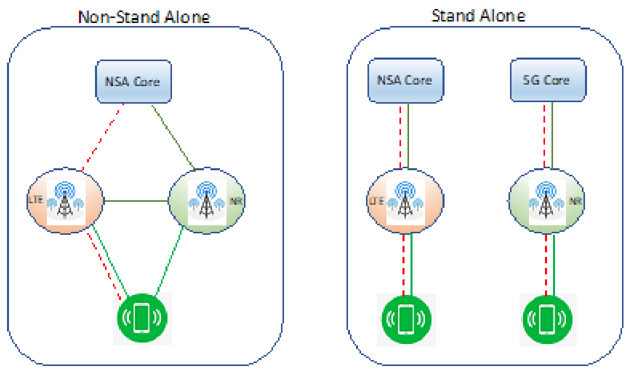


FIGURE 3. 5G NETWORK DEPLOYMENTS, ILLUSTRATING STANDALONE VS. NON-STAND-ALONE (NSA) ARCHITECTURE

5G NETWORK

As the wireless industry is transitioning from 4G LTE to 5G, two solutions defined by 3GPP for 5G networks in Release 15 are shown in Figure 3 and described next:

5G Non-Standalone (NSA): In 5G NSA implementations, the LTE radio access network, and the core network (i.e., the Evolved Packet Core (EPC)) are used as the anchor for mobility management and coverage. New 5G channels, using 5G New Radio (NR) for radio access, provide expanded capacity/throughput to enable 5G enhanced mobile broadband (eMBB) use cases.

- **5G Standalone (SA)**: The 5G SA architecture includes 5G NR for radio access and the 5G Core Network (5GC). 5G SA implementations would support the needs of eMBB, ultra-reliable and low latency communications (URLLC) and massive machine type communications (mMTC) use cases.
- Phase 1B testing discussed in this document has utilized the 5G NSA architecture. Figure 4 shares a high-level architecture diagram for UAS using a 5G NSA network for connectivity.

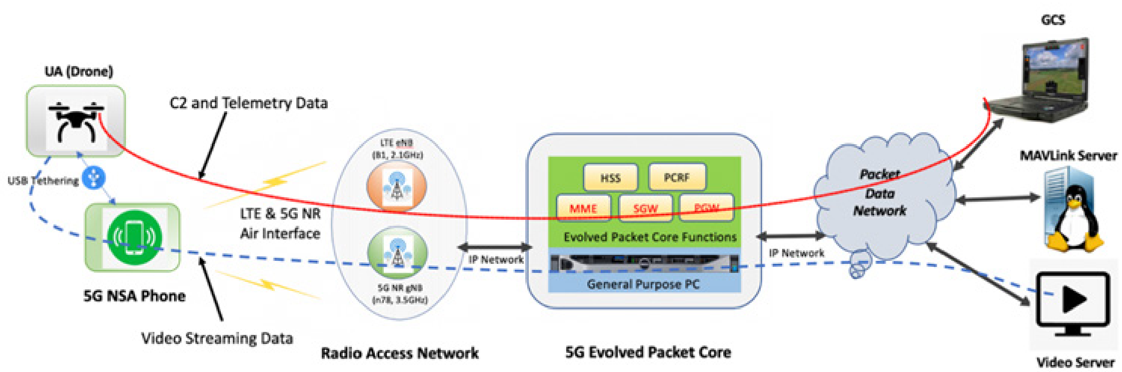


FIGURE 4. UAS OVER 5G NSA NETWORK

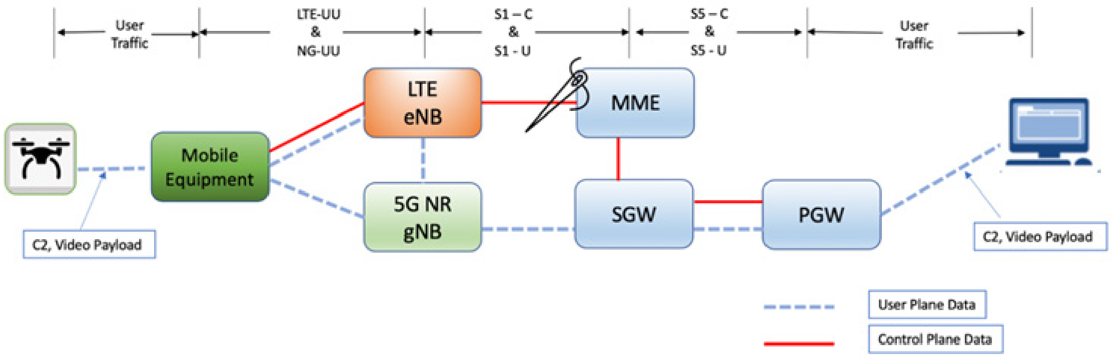


FIGURE 5. DATA SESSION OVER 5G NSA NETWORK

Pseudo Test Algorithm

This section describes the high-level algorithm for Phase 1B drone testing in VT's Drone Park. Figure 6 and Figure 7 show rectangular counterclockwise flight patterns for the drone.

Figure 6 shows an ideal flight pattern that was developed by the VT/CCI team based on their earlier activities at the site and considering prior spectrum readings of wireless activity on LTE Bands 1 and 2 within the VT drone park.

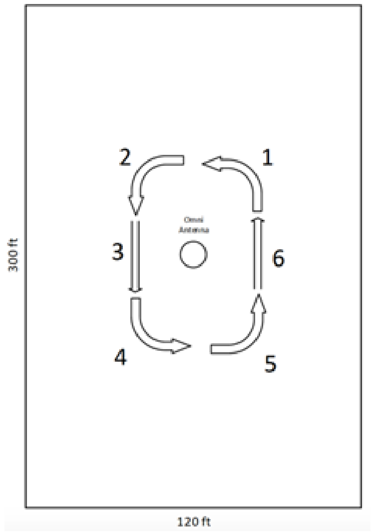


FIGURE 6. IDEAL DRONE FLIGHT PATTERN

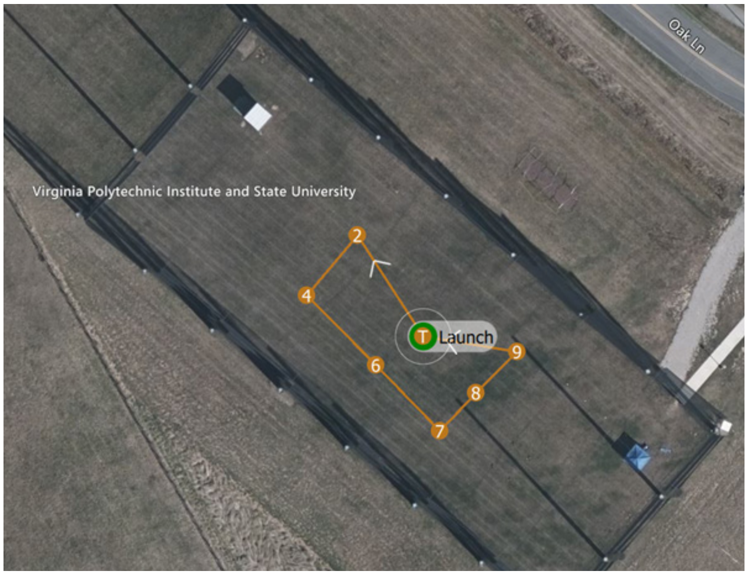


FIGURE 7. ACTUAL FLIGHT PATTERN CREATED IN QGROUNDCONTROL FOR THE EXPERIMENTS

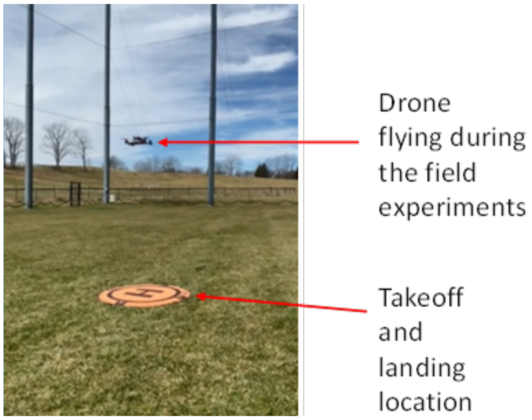


FIGURE 8. DRONE DURING EXPERIMENTS

Figure 7 shows the pattern used in the field, based on the conditions at the time of the experiments. The updated pattern was used to perform the experiments during the flight tests documented in this report. Figure 8 shows the drone airborne during one of our experiments at the Drone Park.

For a given flight pattern and drone altitude, the following data was planned to be collected:

- C2 and Telemetry Data: DL/UL throughput, latency, jitter, and/or Packet Error Rate (PER).
- Video Payload: actual video transmitted from the drone was captured during the test at

the video server. Throughput values were also observed at the Amari Callbox during the experiments.

The drone was envisioned to fly between the various locations and hover at a given location and flight altitude (e.g., z=10ft) for the time needed to run the experiment at that location. The drone would then be commanded to fly to the next ith location, and the process would be repeated.

Figure 9 and Figure 10 illustrate UA exchanging C2 communication data to GCS and Video Streaming to Video server over 5G NSA network established at VT Drone Park.

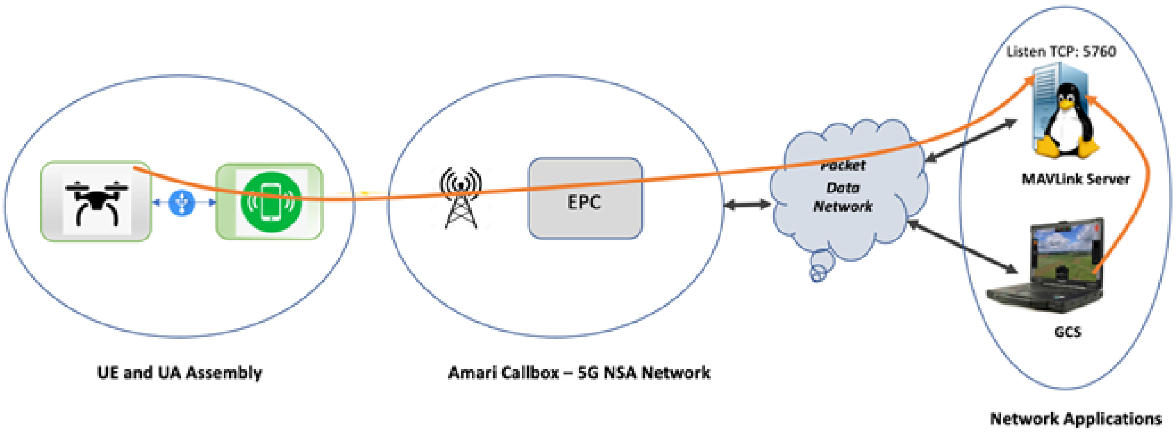


FIGURE 9. C2 COMMUNICATION OVER 5G NSA

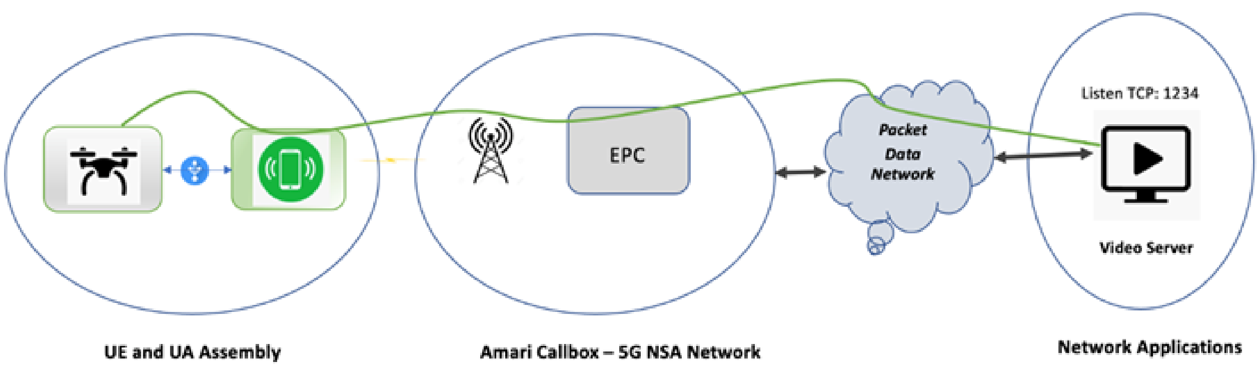


FIGURE 10. VIDEO STREAMING OVER 5G NSA

Test Setup Overview

TABLE 1. TEST SETUP OVERVIEW

Data	Description	
Location	Virginia Tech Drone Park	
Environment	Enclosed facility, within the drone cage (90m x 37m)	
Flight Route	As described in the previous section	
UA Altitudes	Ground, 10 ft, 30 ft AGL	
UA Speed(s)	2 – 5 m/s	
Number of Drones	1	
Frequency Band	B1 band (2.1 GHz) for LTE and n78 band (3.5GHz) for 5G-NR	
Channel BW(s)	LTE: 5MHz and 5GNR: 20 MHz (configurable)	
BS Antenna	Antenna Configuration	MIMO
	Antenna Pattern	Omni
Network Configuration	• PLMN: 001-01 • IMSI: 00101 0000 0000 002 • APN: internet • QCI: 9 • UE IP Address block: 192.168.3.x/24	
Data Type	C2	Command and control from GCS to UA and telemetry data from UA to GCS
	Payload (optional)	Real time streaming video from Drone to server
Test Types	Assess safe operation of UA and communication for UAS echo system over 5G NSA (Option 3X) network while performing actual mission of the operation.	
Data Collection	• UA: On device logging (include battery level) • UE: Android application • 5G Network: Amari Callbox logs	

Data Collection

UA-RELATED METRICS

Telemetry from UA—Capture raw sensor data from UA

- UA Position (Latitude/Longitude)
- UA Altitude(s)
- UA Velocity
- UA Heading
- Update Rate (initially planned to capture updates at 1-2 Hz)
- Timestamp

Other Operational Metrics

- UA Battery Status (Not Applicable)
  The “Cell Info Logger” Android application supported logging battery capacity; however, the UE was tethered via USB to the UA Flight Control Board, which would charge the UE, thus rendered the logging of battery capacity moot.
- UA Flight Time (Not Applicable)
  This parameter can be derived from individual telemetry flight logs and was not explicitly recorded in our data collection.recorded in our data collection.

NETWORK-RELATED METRICS

5G Mobile Device (attached to UA)

- A measure of received signal level
  - DL Reference Signal Received Power (RSRP for LTE or SS RSRP for 5G NR)
- A measure of received signal quality
  - DL Reference Signal Received Quality (RSRQ for LTE or SS RSRQ for 5G NR)

5G Network

- Evolved Packet System Session data[9] S1U Session Data (IP traffic between eNB/gNB and SGW)
- S5U Session Data (IP traffic between SGW and PGW [Packet Data Network Gateway])
- SGI interface stats (IP traffic between PGW and network device)

UE and Network device

- Round Trip Delay (Latency)—Ping test from client to server
- Throughput, Bandwidth, Packet Loss and Jitter—iPerf test from client to server

Initial Configuration and Setup

UAS (Drone and GCS)

DRONE SOFTWARE

Drone Operating System Configuration

The drone ran a special version of Raspbian Linux distribution for the Raspberry Pi (RPi) for the Navio2[10]

The following additional changes should be made to the install:

/etc/network/interfaces : add "eth0"to the end of the line that says "auto lo"

/etc/dhcpcd.conf : add "interface eth0" and "static ip_address=192.168.7.10/24"

/etc/wpa_supplicant/wpa_supplicant.conf: add WiFi networks here like this example:

```
network={
ssid="WiFi AP"
psk="passwd"
}
```

Flight Control Software–ArduPilot arducopter

The ArduPilot ArduCopter flight control software was installed on a Raspberry Pi 4 (RPi) hosting the Raspbian OS with a Navio 2 daughterboard. The Navio 2 daughterboard provided the sensors necessary for stable flight including 9-DOF IMU, barometer, GPS, and motor control. The RPi was configured to start ArduCopter as a service at boot. While ArduCopter was running, it provided telemetry updates using the interface defined in its configuration file, found in /etc/default/arducopter.conf.

For C2, ArduCopter used the MAVLink Protocol. ArduCopter was configured to send MAVLink messages as UDP packets to a specific static IP address. In one configuration, ArduCopter connected to localhost (127.0.0.1) to a collocated mavlink_routerd service. MavLink Router in turn forwarded MAVLink packets to another IP address, such as another mavlink_routerd instance running on a remote machine. This was the configuration used to fly the drone connected through the Internet through a public server running MavLink Router. For the flight test at VT Blacksburg using a private network, we used a similar configuration but used the GCS as the MavLink Server. In this configuration the drone sends the MAVLink messages directly to a mavlink_routerd service running on the GCS computer that was in turn connected to the GCS software.

The settings for the IP address were set in the file /etc/default/arducopter on the Raspberry Pi on the drone. We used the default Ardupilot port of 14550.

We also used the MRobotics.io SiK Telemetry radios as a backup. They appear as device "/dev/ttyAMA0" on the Raspberry Pi.

For example, when the drone was flown using the mavlink_routerd, the command line options would be -A udp:127.0.0.1:14550 -C /dev/ttyAMA0.

The service /etc/systemd/system/arducopter.service automatically started the flight software on boot-up.

Packet forwarder–mavlink_routerd

The daemon mavlink_routerd will be hosted on a machine separate from the UA and act as a kind of "man-in-the-middle", allowing the GCS to connect to the mavlink router server and send/receive MAVLink messages. The main configuration file is in /etc/mavlink-router/main.

conf. When using the drone in this point it is typically configurated similar to this:

```
[UdpEndpoint localhost]
Mode=Eavesdropping
Address = 127.0.0.1
Port = 14550

[TcpEndpoint niles_house]
Addressss = 71.246.207.117
Port = 5760
RetryTimeout=10
```

The service /etc/systemd/system/mavlink-router.service automatically started the mavlink router daemon on startup. It was disabled if mavlink_routerd was not used.

Camera Server–raspivid

By default, the camera server received UDP packets from the drone via port 5000 and either displayed it to the screen or logged it to a file. In this configuration, the drone had to be hard coded with the IP address of the camera server. We used the VideoLAN VLC media player.[11] The drone sent the video using the raspivid program; raspivid sent a raw H.264 stream. The command-line version of VLC was cvlc.

To listen on port 5000 and display to the stream to the screen:

```
cvlc udp://@:5000 --demux=h264
```

To listen on port 5000 and log the output to the file udp_stream.ts

```
cvlc udp://@:5000 --demux=h264 --sout file/ts:udp_stream.ts
```

Future work should improve this camera capability and allow the other clients to connect and receive the stream.

Network Performance–iPerf3 Server (UE Section)

The service /etc/systemd/system/iperf3.service started the "iperf3 -s" server mode on the drone that could be interrogated by various other hosts on the network.

Network Traffic Statistics–logbytes.sh

The service /etc/systemd/system/logbytes.service automatically started the script /home/pi/logbytes.sh shell script on startup. That script printed the seconds since January 1, 1970 (UNIX epoch) and took the "usb0" interface line (cell phone over USB tethering) out of the Linux /proc/net/dev at sent the concatenate the output to the end of the file: /var/log/usb0_bytes.log at a rate of one new entry line per second. The file grows unbounded, so periodic maintenance to clear the log file was required.

Drone Remote Access–Management

The drone was set up to DHCP on any known Wi-Fi connection and had a static IP address of 192.168.0.10 on the Ethernet port. This allowed the GCS or other test computer to connect to the drone and configure it.

GCS SOFTWARE

QGround Control

This is the MAVLink based ground control software used for the experiment, GUI display as shown in Figure 10 below. On a private network, no additional configuration was needed and the drone automatically connected to it.

For an internet-based connection, you needed to connect it via the gear's icon to the TCP port of the IP address of the server running MAVLink router, as described in Section 3.1.1.2.

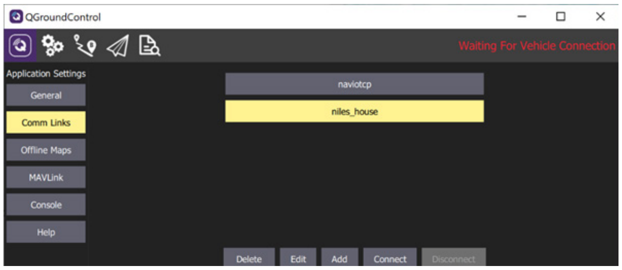


FIGURE 11. QGROUND CONTROL

Network Performance-iPerf3 Client

Sending data and testing performance (throughput, jitter, packet loss).

/usr/local/bin/iperf3 was the path to the program to test Bandwidth and Jitter. Output was logged to a file for later analysis.

Examples:

iperf3 -c 192.168.1.184 (will run a TCP test for 10 seconds if the drone is at 192.168.1.184)

iperf3 -u -c 192.168.1.184 (will run a UDP test for 10 seconds)

Network Traffic Statistics (MacOS version)–logbytes.sh

The following script was used to capture network traffic statistics on the GCS. There was no /proc file system so "netstat" had to be used instead:

```
#!/bin/sh
IFS=' '
OUTFILE=~/GCS_bytes.log
while true; do
 DATE=`date +%s`
 OUTPUT=`netstat -s 2> /dev/null | grep byte | grep IPv4 | grep datagrams | sed 's/(//g' | tr '\n' ' '`
 read -ra ADDR <<< "$OUTPUT"
 echo $DATE "${ADDR[0]}" "${ADDR[2]}" "${ADDR[6]}" "${ADDR[8]}" | tr '\t' ' ' >> $OUTFILE
 sleep 1;
done
```

Note: This script had to be started manually on the GCS before testing began

Camera Player–web browser

The use of a cell phone as an interface for cellular communication created a "firewall" between the phone and the RPi. This "firewall" prevented the UA from acting as server, such as hosting a web server that streamed video. In the Phase 1B configuration, the UA always acted as a client that connected to a server, such as the mavlink_routerd server.

5G RAN and EPC (Amari Callbox)

5G EPC & RAN for the experiment was 3GPP Release 15 compliant with Option-3-enabled, supporting NR and LTE accesses. The 5G EPC is implicitly backward compatible and fully interworking with legacy LTE RAN.

- Option 3X: Dual connectivity NR/LTE: UE anchored to LTE radio and 5G EPC and used LTE and NR for user plane traffic. Data traffic was aggregated in the gNB.
- LTE-only UE: It was accepted into the Option-3-enabled EPC and RAN architecture for LTE access only.

FIGURE 12. AMARI CALLBOX AT VT DRONE PARK

For the RAN and Core part, the Amari Callbox from AmariSoft, a proprietary 5G-In-A-Box was used. The RAN and Core were configured with the 5G NSA Option 3X. The Amari[12] Callbox Pro[13] product was used in the tests described in this document.

The Amari Callbox baseline configuration used during testing was derived by CCI/VT team through previous experimentation to optimize performance. Figure 12 shows the Amari Callbox as part of the setup at the VT Drone Park during the experiments.

RADIO NETWORK (RF CONNECTIONS)

In Figure 13, blue indicates channel 1 and purple indicates channel 2.

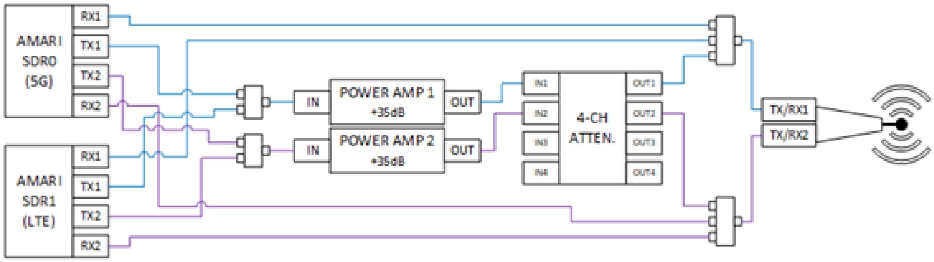


FIGURE 13. AMARI CALLBOX RADIO CONFIGURATION

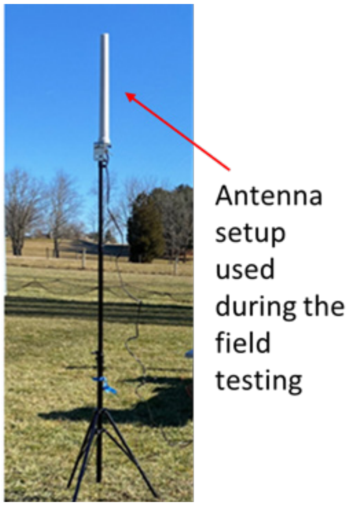


FIGURE 14. OMNI ANTENNA AT VT DRONE PARK

RF components were added to the front end of the Amari Callbox Pro to enable mobility, supplying power amplification, and omni-directional RF emission. Front-end amplification was illustrated with the outdoor system architecture shown. The Amari Callbox Pro front end was represented with two of its contained software-defined radios. By utilizing SDR0 for 5G NR and SDR1 for LTE, the architecture enabled both NSA and SA 5G NR. Both Frequency Division Duplex (FDD) and Time Division Duplex (TDD) were enabled with this architecture. Each power amplifier provides a gain of 35dB. A multi-channel programmable attenuator was used to control output RF power into the attached antenna. Attenuation was not used during this testing. An omni-directional antenna was connected to the channel 1 and 2 combiners. The omnidirectional antenna provided a maximum gain of 6.2 dBi, however for the 5G NR band of interest (n78, 3550 MHz) it provided 2 dBi gain. The antenna model used was OMNI-600-02.[14] Figure 14 shows the antenna setup used during testing.

IP Network

The IP network architecture consists of a private network comprised of the Amari Callbox Pro, a laptop that controlled the Amari Callbox Pro, and a second laptop running GCS Software that enabled sending and receiving commands for the Raspberry Pi. The Amari Callbox Pro communicated with a connected 5G UE (Google Pixel 5). The Raspberry Pi on the drone tethered to the 5G UE, enabling an end-to-end communications link from the GCS to the drone. Figure 15 shows the high-level architecture.

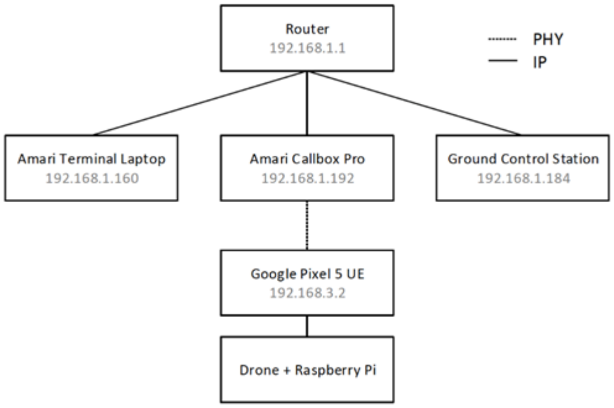


FIGURE 15. HIGH LEVEL IP NETWORK ARCHITECTURE

Mobile Device and Drone Integration

The integration of mobile device and drone is illustrated in Figure 16.

Following elements are included:

- User Equipment: Google Pixel 5 (5G NSA phone) supporting USB tethering
- Drone: Ardupilot flight software running on Raspberry-Pi
- UAS Ground Control System–Open-Source GCS software (QGroundControl) running on MITRE Engenuity MacBook Air laptop

FIGURE 16. DRONE ASSEMBLY WITH MOBILE DEVICE

List of Materials

MITRE TEAM

Drone Assembly

- Lynx motion HQuad500 Frame
- Frisky Taranis X9D Plus SE 2019 RC Transmitter
- Pixel 5 cellphone
- USB-C to USB-A 6" cable for cellphone
- LiPO 3S battery
- 900 MHz Telemetry Antenna

GCS Computer

- Hardware
  - Laptop
  - Ethernet dongle
  - RJ45 Cat5/6 twisted pair ethernet cord (to connect GCS)
  - U.S. Type A plug to USB-c AC/DC Converter (Power adapter)
  - USB-c to USB-c 6-foot cord
  - 900 MHz Telemetry Antenna (backup C2)
  - Mini-USB to USB-A cord to charge transmitter
  - USB-C to USB-A for Telemetry Antenna
- Software
  - QGroundControl
  - iPerf
  - Bash script for recording packet counts (see "logbytes.sh")

Additional Materials

- Additional LiPO Batteries (currently four Zeee 5200mAh 3S)
- LiPO firesafe storage bag
- LiPO Charger: EV-Peak C1-XR
- Various zip-ties to secure phone and other elements
- Sticky back foam padding material (to secure elements and provide vibration damping)
- Double-Lock Velcro
- Duct Tape
- Microphone muffs (to protect barometer)
- Additional Propellers (including extras) [9" diameter]
- Tools:
  - 8 mm nut driver (to secure propellers)
  - Small Philips screwdriver (for various disassembly and repair)
  - 3/32" hex driver (for disassembly and repair)
  - Metric hex set
- Additional miscellaneous spare screws and standoffs
- Camera
- Tripod
- Extra drone issue debug items:
  - Portable USB keyboard
  - Portable HDMI screen
  - Micro HDMI to HDMI cable for Raspberry Pi 4
  - Raspberry Pi 4 USB-C power supply

CCI/VT TEAM

- The CCI 5G End-to-End NSA architecture included the following components.

Long Case

Specific details of Omni antenna and Power Amplifier are described below.

- Power Amplifier ZHL-42W+ Gain Block, 10 – 4200 MHz, 50Ω Connector Type: SMA
- Omni Antenna 600-02: Freq Bands: 3300 MHz – 3800 MHz Radiated Power: 23 dBm EIRP: 30.7 dBm
- Mean/Peak Gain: 2/6.2 dB Freq. Tolerance: 5 MHz Station Class: Fixed

TABLE 2. LONG CASE

Item	Quanity
Omni Antenna	2
Rubber Straps	4
DC Power Supply	1
Mouse	1
Banana Clip Sets	2
Raspberry Pi & Cables	1
Zip Ties	1
Rubber Bands	1
Duct Tape	1
USB Drive	1

TABLE 3. AMARI CALLBOX CASE

Item	Quanity
Amari Callbox	1
10-Port Switch	1
5-Port Switch	1
Power Adapter	1

TABLE 4. RF CASE

Item	Quanity
Directional Antenna	2
Power Amplifier	2
20 ft RF Cables (SMA)	4
Short SMA Cables	4
Attenuator & Cable	1
SMA Wrench	2
Hotspot & Power Cable	1
Phones and Chargers	4
Turntable Ethernet & Power	1

TABLE 5. POWER CASE

Item	Quanity
Inverter	1
Inverter Cables	1
Power Strip	1
Extension Cord 100 ft	1
Extension Cord 200 ft	1
Wrench	2
Pliers	1
Scissors	4
Screwdriver Set	1
Clear Tape	1
Ethernet Cables	8

TABLE 6. BACKPACK

Item	Quanity
Portable Monitor	1
Laptop	1
Test Phones	4
Test Phone Chargers	4
Laptop Adapter	1

TABLE 7. STANDALONE

Item	Quanity
Sandbags	4
Antenna Stands	4
Wagons	1
Field Fox Spectrum Analyzer	1

Data Collection–Log Files

MAVLINK BINARY LOGS–DRONE

Figure 17 demonstrates how it was verified that QGroundControl had enabled logging in the "Log Download" section.

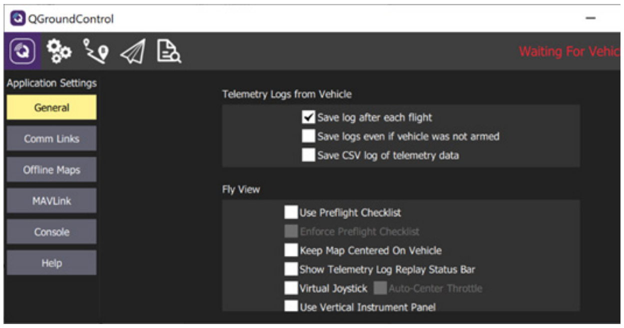


FIGURE 17. LOG DOWNLOAD SECTION

LOGBYTES–DRONE

The file usb0_logbytes.log contained counters of the various bytes received and transmitted from the drone.

LOGBYTES–GCS

The file GCS_logbytes.log contained counters of the bytes and packets received and transmitted by the GCS.

The first field came from the date command. The remaining fields came from the Linux kernel.

```
Time in seconds since Jan 1, 1970.stats->rx_bytes,
stats->rx_packets,
stats->rx_errors,
stats->rx_dropped
stats->rx_missed_errors,
stats->rx_fifo_errors,
stats->rx_length_errors
stats->rx_over_errors +
stats->rx_crc_errors +
stats->rx_frame_errors,
stats->rx_compressed,
stats->multicast,
stats->tx_bytes,
stats->tx_packets,
stats->tx_errors,
stats->tx_dropped,
stats->tx_fifo_errors,
stats->collisions,
stats->tx_carrier_errors
stats->tx_aborted_errors
stats->tx_window_errors
stats->tx_heartbeat_errors,
stats->tx_compressed;
```

Example output:

```
1612972456 165834166 147098 0 0 0 0 0 0 10197024 54411 0 0 0 0 0 0
1612972457 165835862 147106 0 0 0 0 0 0 10201452 54421 0 0 0 0 0 0
1612972458 165843942 147125 0 0 0 0 0 0 10208737 54452 0 0 0 0 0 0
1612972459 165845328 147129 0 0 0 0 0 0 10208869 54454 0 0 0 0 0 0
1612972460 165845328 147129 0 0 0 0 0 0 10208869 54454 0 0 0 0 0 0
1612972461 165845328 147129 0 0 0 0 0 0 10208869 54454 0 0 0 0 0 0
1612972462 165845328 147129 0 0 0 0 0 0 10208869 54454 0 0 0 0 0 0
1612972463 165845328 147129 0 0 0 0 0 0 10208869 54454 0 0 0 0 0 0
1612972464 165845328 147129 0 0 0 0 0 0 10209083 54455 0 0 0 0 0 0
1612972465 165845328 147129 0 0 0 0 0 0 10209297 54456 0 0 0 0 0 0
1612972466 165855018 147156 0 0 0 0 0 0 10217584 54498 0 0 0 0 0 0
1612972467 165855018 147156 0 0 0 0 0 0 10217798 54499 0 0 0 0 0 0
1612972468 165864484 147179 0 0 0 0 0 0 10225103 54532 0 0 0 0 0 0
1612972469 165864484 147179 0 0 0 0 0 0 10225103 54532 0 0 0 0 0 0
1612972470 165864536 147180 0 0 0 0 0 0 10225197 54533 0 0 0 0 0 0
```

ANDROID LOGGING APP–UE

The phone was installed with a custom Android application called "Cell Info Logger" (CIL) and the app captured RF and telephony metrics as seen by the phone. These key/value pairs were logged as JSON strings to a local file using the following naming convention: allcellinfo_YYYY-MM-DD_HH-MM.raw. The log file was accessible using the "Files" app, which came pre-loaded on the phone, under Internal Storage > Documents > YYYY-MM-DD. The app also captured GPS latitude, longitude, and altitude. The raw data file was converted to machine-readable JSON format by using the "cvt2csv.py" utility.

The JSON data file consists of several data types, which are described in Figure 18.

FIGURE 18 JSON DATA FILE

The tree structure shown above was the default for any record stored in the data file. The parent key was "allcellinfo," which consists of a list of elements, denoted by the "0" value shown in the figure. Each element consists of two common keys, the "timestamp" and the "record." The "timestamp" represented the UNIX Epoch time in milliseconds. The time value could be converted to a human readable date using common date/time packages found in most high-level languages. The "record" key stored the captured data element, each of which is discussed below. From the schema above, we can see how to access each record in the data file.

FIGURE 19. JSON SCHEMAS WITH SAMPLE DATA FOR "CELLSIGNALSTRENGTHNR"AND "GPS" RECORDS

On the top of Figure 19, we show the schema for the "cellsignalstrengthnr" record, which was captured when the phone was the presence of a 5G-enabled tower. Note that unreported values were assigned the value of "na." More information on the definition of each key/value pair can be found in Android API documentation.[15]

On the bottom of Figure 19, we show the schema for the "gps" record, which captured the phone's position in a WGS84 coordinate reference system. More information on the definition of each key/value pair can be found in Android API documentation.[16]

The most common data types were the "CellIdentityLte" and "CellSignalStrengthLte" records. Figure 20 provides some sample fields that been collected. More information on the definition of each key/value pair can be found in Android API documentation.[17]

The next common data type was captured by the service state of the phone, shown on the left in. The schema for the service state was complex and contained hidden data types not found in the standard documentation. You can find more information on the key/value pairs in Android API documentation.[18]

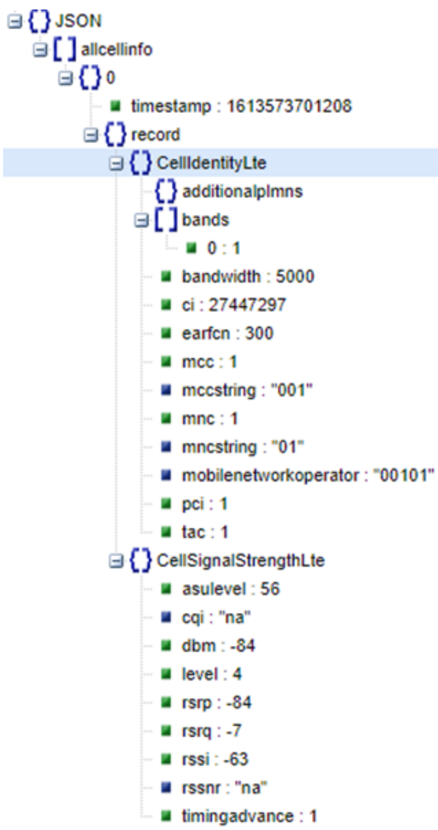
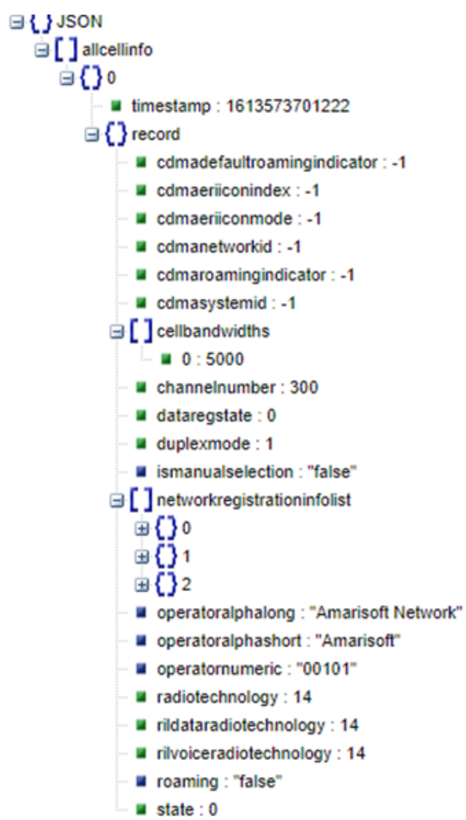


FIGURE 20. JJSON SCHEMAS WITH SAMPLE DATA FOR "CELLIDENTITYLTE" AND"CELLSIGNALSTRENGTHLTE" RECORDS



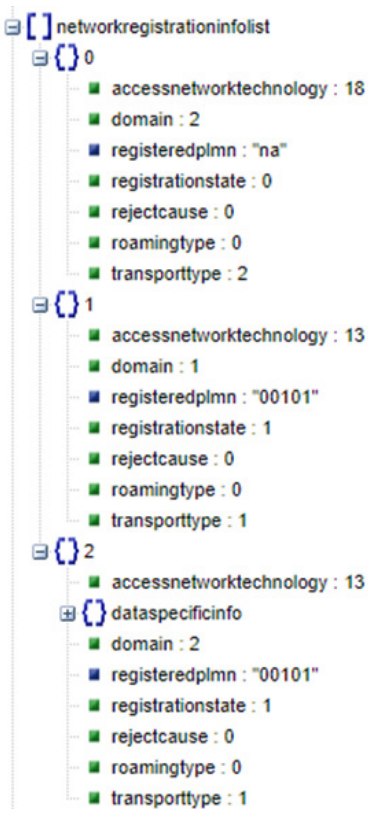


FIGURE 21. JSON SCHEMA WITH SAMPLE DATA AND DETAILS FOR "NETWORKREGISTRATIONINFOLIST"

From Figure 21 we can see that the service state revealed the network operator, bandwidth, and channel number. The "networkregistrationinfolist" (shown on the right) typically consist of an array of length three. More information on the key/value pairs can be found in Android API documentation.[19]

FIGURE 22. FURTHER DETAILS AND SAMPLE DATA FOR "NETWORKREGISTRATIONINFOLIST"

As shown in Figure 22, within the "networkregistrationinfo" there is a data type called "dataspecificinfo." Here it could be determined if the phone was in the presence of a 5G-enabled tower and if the tower was using carrier aggregation.

RAN and Core Network–Amari Callbox

The Amari Callbox generated single log file that included data from RAN and EPC including throughput, packet rate, error rate, and retransmission rate for both LTE and NR.

3.6.4.2 Latency, Throughput, Packet loss, Bandwidth–Laptop (iPerf, ping)

iPerf and ping tests were logged to a file with a time stamp to measure network performance. While the iPerf records were captured completely at the GCS, the record was only timestamped once, upon starting the iPerf server. Subsequent timestamps were not captured by the shell script. Sample data is shown in Figure 23.

```
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
Accepted connection from              , port 44462
[ 5] local              port 5201 connected to              port 44464
[ ID] Interval Transfer Bitrate
[ 5] 0.00-1.00 sec 2.99 MBytes 25.1 Mbits/sec
[ 5] 1.00-2.00 sec 4.37 MBytes 36.7 Mbits/sec
[ 5] 2.00-3.00 sec 3.48 MBytes 29.2 Mbits/sec
[ 5] 3.00-4.00 sec 2.92 MBytes 24.5 Mbits/sec
[ 5] 4.00-5.00 sec 3.34 MBytes 28.0 Mbits/sec
[ 5] 5.00-6.00 sec 3.71 MBytes 31.1 Mbits/sec
[ 5] 6.00-7.00 sec 3.29 MBytes 27.6 Mbits/sec
[ 5] 7.00-8.00 sec 4.15 MBytes 34.8 Mbits/sec
[ 5] 8.00-9.00 sec 3.33 MBytes 27.9 Mbits/sec
[ 5] 9.00-10.00 sec 4.47 MBytes 37.5 Mbits/sec
[ 5] 10.00-10.03 sec 177 KBytes 43.6 Mbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval Transfer Bitrate
[ 5] 0.00-10.03 sec 36.2 MBytes 30.3 Mbits/sec receiver
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
Accepted connection from              , port 44466
[ 5] local              port 5201 connected to              port 44468
[ ID] Interval Transfer Bitrate
[ 5] 0.00-1.00 sec 3.53 MBytes 29.6 Mbits/sec
[ 5] 1.00-2.00 sec 4.06 MBytes 34.0 Mbits/sec
[ 5] 2.00-3.00 sec 3.94 MBytes 33.0 Mbits/sec
[ 5] 3.00-4.00 sec 3.87 MBytes 32.4 Mbits/sec
[ 5] 4.00-5.00 sec 3.48 MBytes 29.2 Mbits/sec
[ 5] 5.00-6.00 sec 3.84 MBytes 32.2 Mbits/sec
[ 5] 6.00-7.00 sec 4.46 MBytes 37.4 Mbits/sec
[ 5] 7.00-8.00 sec 4.07 MBytes 34.2 Mbits/sec
[ 5] 8.00-9.00 sec 4.02 MBytes 33.7 Mbits/sec
[ 5] 9.00-10.00 sec 3.80 MBytes 31.9 Mbits/sec
[ 5] 10.00-10.06 sec 116 KBytes 16.4 Mbits/sec
- - - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval Transfer Bitrate
[ 5] 0.00-10.06 sec 39.2 MBytes 32.7 Mbits/sec receiver
```

FIGURE 23. SAMPLE IPERF RECORD

Experiments

In this section, each test is described using five sub-sections.

- The test objective outlines the goals of the test.
- Pre-conditions specify any conditions required for the test to be successful.
- Test configuration, if applicable, is any specific configuration that was modified over what was performed in Section 3 for this specific test.
- The test procedure outlines the steps that had to be performed to conduct the test.
- The results sub-section describes the measurement results that were obtained after processing the collected data.

In-Place Test

This test ran the drone UE in a stationary/in-place position (propeller spinning, remaining on the ground).

TEST OBJECTIVE

With the Amari Callbox's Antenna located within the Drone Park the drone was placed on the ground at each of the locations in the measurement set. The measurement locations are shown in Figure 7. The objective was to capture a set of measurements on the ground before the drone executed the flight tests.

PRE-CONDITIONS

The drone and GCS hardware and software setup were completed successfully, as described previously. The Amari Callbox (that provided an end-to-end 5G NSA connectivity) described previously had also been set up successfully. In addition, successful connectivity between the Amari Callbox and the drone existed for both the LTE and 5G NR signals (where LTE cell operated on Band 1 and the 5G cell operated on band n78 as described in the Federal Communications Commission's experiment web portal).

TEST CONFIGURATION

No specific configuration changes were needed for the in-place test.

TEST PROCEDURE

Drone on the ground was placed at each of the locations in the measurement set {P1, P2, … Pn}.

For each location Pi in the measurement set, the following steps were performed:

- Step 1: At location Pi, collected RF link performance metrics at the drone UE using Android app "Cell Info Logger".
  - The RF metrics set was as captured by the Google API on the UE (Android-based phone).
    - Metrics should have included RSRP, RSRQ for LTE
    - Metrics should have included SS-RSRP, SS-RSRQ for 5G NR
- Step 2: At location Pi, ran ping to collect latency samples.
- Step 3: At location Pi, ran the logbytes.sh script at the drone to collect data traffic statistics, logged the number of bytes sent and received by the drone, time-stamped.
- Step 4: At location Pi, collected C2 statistics. Telemetry was sent from the drone and commands were received at the drone (as sent from GCS).

- Step 5a: At location Pi, "./iperf3.sh" needed to be running to capture iPerf3 data. This was in addition to telemetry that continued to be sent from the drone.
- Step 5b: At location Pi, "./receive_video.sh" needed to be running to capture video. This was in addition to telemetry that continued to be sent from the drone.

For these experiments, iPerf3 data or video represented payload data.

The drone was then placed on the ground at next location in the measurement set. Data collection was performed at this location, by repeating the previously described steps.

In parallel to data collection at drone, during the entire time data was collected at the drone for all locations around the Drone Park, the following data was also collected:

- GCS
  - Collected corresponding C2 statistics as seen by GCS during the time only C2 was sent.
- Transmitted commands
- Received telemetry
  - Collected overall data traffic statistics at GCS using the corresponding logbytes.sh script at GCS.
- Video received from drone
- Amari Callbox
  - Generated logs from all layers (from physical layer to IP level).
  - eNodeB and MME generated log files under /tmp/ directory (enb0.log, ims.log and mme.log) that could have been used for further analysis and debugging.

The LTE web interface allowed to analyze Amarisoft LTE software logs and get real time information from the system. Note: After the drone was placed at a given test location on the ground, the signal path from Amari Callbox to drone should not be obstructed by people or equipment.

RESULTS

After processing the data collected at the drone, the GCS and the Amari Callbox, measurement results should have included the list below:

- RF performance statistics for metrics including RSRP, RSRQ as captured at drone UE, with drone at ground level receiving LTE signals
  - o Histograms/PDFs/CDFs for each collected metric
- RF performance statistics for metrics including SS-RSRP, SS-RSRQ, as captured at drone UE, with drone at ground level receiving 5G NR signals
  - o Histograms/PDFs/CDFs for each collected metric
- Results for throughput/data rates for DL and UL for the list below.
  - o C2 and iPerf3
  - o C2 and video
- Results for latency for DL and UL for the list below.
  - C2 and iPerf3
  - C2 and video
- RF performance metrics and other metrics as logged at Amari Callbox

In-Flight Test

TEST OBJECTIVE

With the Amari Callbox's Antenna located within the Drone Park, the drone was expected to fly to a series of test point locations. These locations were the same as the ones used in the In-Place Test, and were denoted as {P1, P2, … Pn}. For the results presented in Section 5, the Amari Callbox's Antenna location was shown with a black circle in the plots included as part of Table 9, as an example.

Two altitudes were planned 10 ft. and 30 ft AGL. At each altitude, the drone was planned to fly around the Drone Park to the series of test point locations described above. The objective was to capture the KPIs that were used for the In-Place Test.

PRE-CONDITIONS

The drone and GCS hardware and software setup had been completed successfully, as described in Section 3.1 of this document. The Amari Callbox had also been set up successfully. In addition, successful connectivity between the Amari Callbox and the drone existed for both the LTE and 5G NR signals.

Furthermore, it was assumed that the In-Place Test was also performed as discussed in the In-Place Test section.

Moreover, it needed to be ensured that the app to collect RF data on drone was running correctly and that all scripts were running correctly to be able to collect measurements at drone, at GCS, and at the Amari callbox.

TEST CONFIGURATION

No specific configuration changes were needed for the in-flight test.

TEST PROCEDURE

Ideally, for a given flight altitude, the drone was flown to each of these locations {P1, P2, … Pn}. It was controlled in "guided mode" where the GCS commanded the drone to each new location. Alternatively, it could have been done in "auto mode" where the drone flew to each location, hovered for a fixed amount of time, and proceeded automatically to the next location; or if the "guided mode" or "auto mode" were not available, the remote pilot could have flown the drone to each test location manually in "loiter mode" or "stabilize mode."

The drone performed tests at each one of the test point locations {P1, P2, … Pn} around the Drone Park.

With the drone in-flight at first altitude (10 ft. AGL), commanded the drone to fly to each of the test point locations.

For each location Pi in the measurement set, the following steps were performed:

- Step 1: At location Pi, collected RF link performance metrics at the drone UE using Android app "Cell Info Logger".
    - The RF metrics set is as captured by the Google API on the UE (Android-based phone).
        - Metrics included RSRP, RSRQ for LTE
        - Metrics included SS-RSRP, SS-RSRQ for 5G NR
- Step 2: At location Pi, ran ping to collect latency samples.

- Step 3: At location Pi, ran the logbytes.sh script at the drone and at GCS to collect data traffic statistics, logging the number of bytes sent and received by the drone, timestamped, and the number of bytes sent and received by GCS also timestamped.
- Step 4: At location Pi, collected C2 statistics. Telemetry was sent from the drone and commands were received at the drone (as sent from GCS).
- Step 5a: At location Pi, "./iperf3.sh" needed to be running to capture iPerf3 data. This was in addition to telemetry that continued to be sent from the drone.
- Step 5b: At location Pi, "./receive_video.sh" needed to be running to capture video. This was in addition to telemetry that continued to be sent from the drone.

For these experiments, iPerf3 data or video represented payload data.

The drone was then commanded to fly to the next location, and the previously mentioned steps were repeated. After collecting data at all points Pi, at this flight altitude, the drone was commanded to land.

During the entire time the drone was collecting data for a given in-flight altitude, the following data was also collected on the ground:

- GCS
  - Collected corresponding C2 statistics as Transmitted commands
- Received telemetry
  - Collected overall data traffic statistics at GCS using the corresponding logbytes.sh script at GCS.
- Video received from drone
- Amari Callbox data:
  - Logs were collected and written in files

Repeated A) for another in-flight altitude. (Note: Due to setup constraints, only 10 ft. altitude flights were executed).

RESULTS

For tests performed at each of the UA flight altitudes:

After processing of data collection at the drone, the GCS, and the Amari Callbox, for each of the UA altitudes, measurement results included the list below:

- RF performance statistics for metrics including RSRP, RSRQ as captured at drone UE, with drone at that given UA altitude receiving LTE signals.
  - Histograms/PDFs/CDFs for each collected metric
- RF performance statistics for metrics including SS-RSRP, SS-RSRQ, as captured at drone UE, with drone at that given UA altitude receiving 5G NR signals.
  - o Histograms/PDFs/CDFs for each collected metric
- Results for throughput/data rates for DL and UL for the list below.
  - o C2 and iPerf3
  - o C2 and video
- Results for latency for DL and UL for the list below.
  - o C2 and iPerf3
  - o C2 and video
- RF performance metrics and other metrics as logged at Amari Callbox.

Data Post Processing

This section presents the metrics gathered on the Amari callbox during the March 2nd to 5th, 2021 network deployment and experiment execution. The data shown here corresponded to the average data collected over the entire duration of the network deployment at the VT Drone Park including network setup, network testing and drone flight time.

Figure 24 shows the Physical Random-Access Channel (PRACH) SNR. The data shown here was gathered before the data transmission started. The results show that the LTE cell had a SNR of approximately 25dB, while the NR SNR was about 10dB. This is likely a contributing factor explaining why the LTE cells were the preferred over the NR cells on the user equipment on the aircraft.

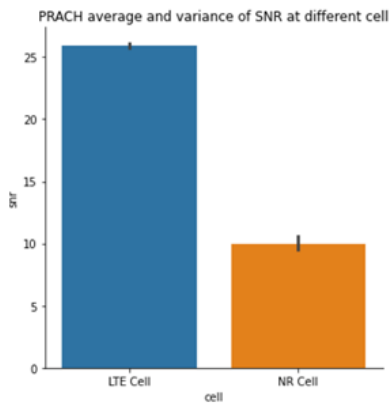


FIGURE 24. PRACH SNR

Figure 25 compares the SNR during the data transmission. We noted that when the NR cell was connected, the average SNR was higher than for the LTE cell. Note that when the NR cell was used its SNR was higher than that of LTE cell.

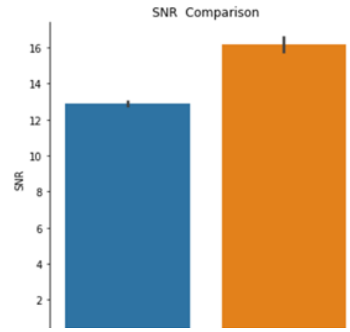


FIGURE 25. SNR COMPARISON DURING DATA TRANSMISSION

Figure 26 compares the LTE and NR cell throughput performance for both downlink and uplink. For the experiment purpose, the focus was uplink, i.e. from UE to base station. We can see that when NR was connected, the uplink throughput performance was significantly higher than the LTE cell, although most of the time LTE cell was selected by the UE during the experiments.

The Channel Quality Indicator (CQI) and throughput regression is shown as Figure 27 for LTE and Figure 28 for NR. In the regression plots, for the NR cells, throughput and CQI had positive correlation, but LTE did not. This generally indicates poor antenna coverage during the UA flight. The LTE cell serves as the default cell for connections when not in an ideal channel environment. To fully understand this behavior, further investigation would be required.

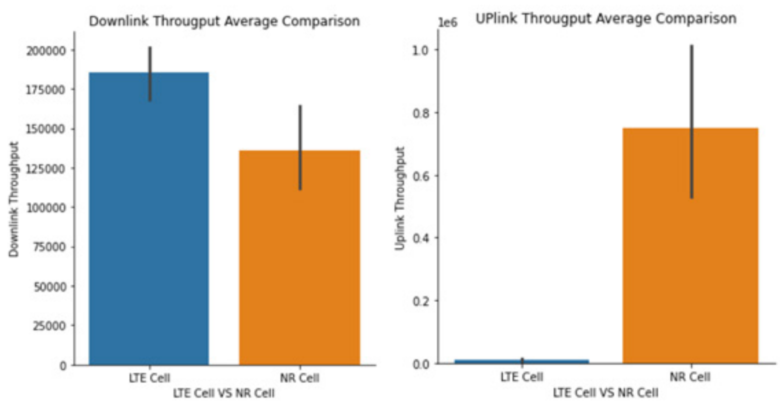


FIGURE 26 LTE AND NR CELL THROUGHPUT PERFORMANCE

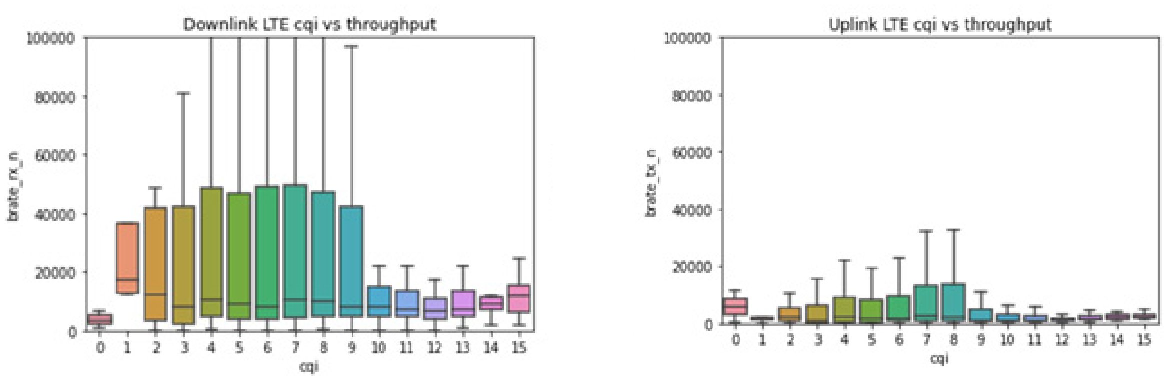


FIGURE 27. LTE: CQI AND THROUGHPUT

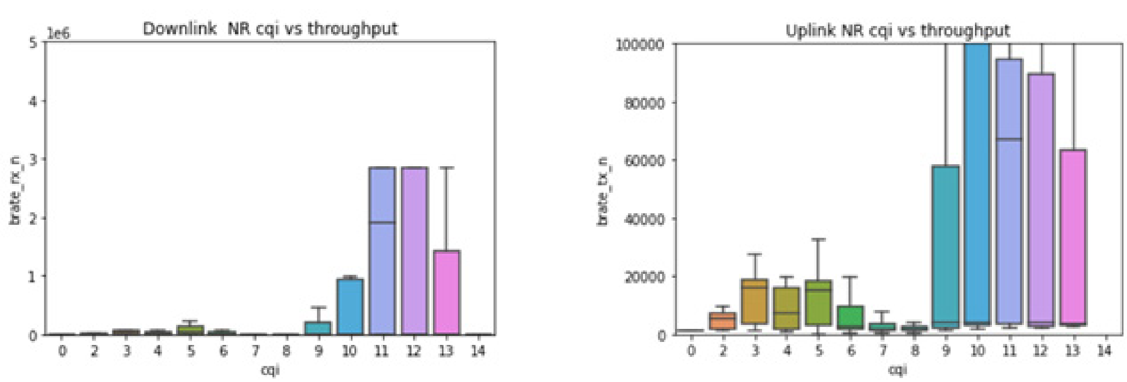


FIGURE 28. NR: CQI AND THROUGHPUT

timestamp_sec	Lat	Lng	RelHomeAlt	path	bands	bandwidth	earfcn	mobilenetworko	tac	rsrp	rsrq	rssi	rssnr	timingadvance	ssrsrp	ssrsrq	sssinr	rxbytes	rxpackets	txbytes	txpackets
1614780601	37.2230298	-80.4325693	0	0	[1]	5000	300	101	1	-81	-8	-81		1				1247823	2828	388830	3496
1614780601	37.2230298	-80.4325693	0	0											-100	-11		1247823	2828	388830	3496
1614780601	37.2230298	-80.4325695	-0.01	0	[1]	5000	300	101	1	-81	-8	-81		1				1247823	2828	388830	3496
1614780601	37.2230298	-80.4325695	-0.01	0											-100	-11		1247823	2828	388830	3496
1614780601	37.22303	-80.4325696	-0.03	0	[1]	5000	300	101	1	-81	-8	-81		1				1247823	2828	388830	3496
1614780601	37.22303	-80.4325696	-0.03	0											-100	-11		1247823	2828	388830	3496
1614780601	37.2230301	-80.4325698	-0.05	0	[1]	5000	300	101	1	-81	-8	-81		1				1247823	2828	388830	3496
1614780601	37.2230301	-80.4325698	-0.05	0											-100	-11		1247823	2828	388830	3496
1614780602	37.2230302	-80.4325699	-0.07	0											-100	-11		1263369	2851	391571	3515
1614780602	37.2230302	-80.4325699	-0.07	0											-100	-11		1263369	2851	391571	3515
1614780602	37.2230301	-80.4325701	-0.09	0											-100	-11		1263369	2851	391571	3515
1614780602	37.2230301	-80.4325701	-0.09	0											-100	-11		1263369	2851	391571	3515
1614780602	37.22303	-80.4325702	-0.11	0											-100	-11		1263369	2851	391571	3515
1614780602	37.22303	-80.4325702	-0.11	0											-100	-11		1263369	2851	391571	3515
1614780602	37.2230298	-80.4325703	-0.13	0											-100	-11		1263369	2851	391571	3515
1614780602	37.2230298	-80.4325703	-0.13	0											-100	-11		1263369	2851	391571	3515
1614780602	37.2230298	-80.4325704	-0.15	0											-100	-11		1263369	2851	391571	3515
1614780602	37.2230298	-80.4325704	-0.15	0											-100	-11		1263369	2851	391571	3515
1614780602	37.2230298	-80.4325705	-0.17	0											-100	-11		1263369	2851	391571	3515

FIGURE 29. MERGED SAMPLE DATA

A note on the collection of In-Place data, in which the vehicle was purposefully placed on the ground at specific locations. During post-processing, we realized that the UA had not captured telemetry data corresponding to our ground positions. This was due to the default logging configuration onboard the UA, which only enabled logging of telemetry data when the UA flight controller was active.

ACTUAL FLIGHT PATTERNS

Table 8 shows the actual flight patterns from March 3. The flight patterns were different every time and altitude varied significantly because the drone was operated under manual mode.

TABLE 8. ACTUAL FLIGHT PATTERNS—MARCH 3

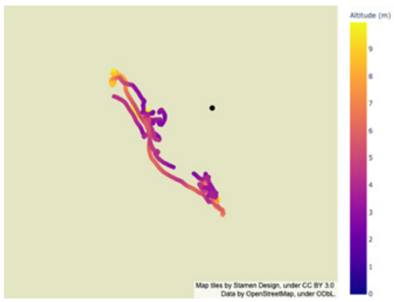
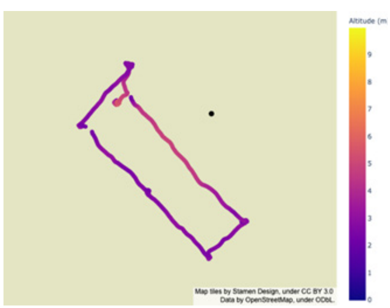
1	2	3
		
4	5	6
		

Table 9 shows the actual flight patterns from March 4. Compared to flight patterns on March 3, we observed that the flight patterns were more consistent when operating under guided mode. GCS commanded the drone to proceed to each waypoint in real time over cellular connection. Drone successfully flew at the commanded altitude of 10 ft.

TABLE 9. MARCH 4 DRONE FLIGHT PATTERNS, OPERATED UNDER GUIDED MODE

1	2	3
		
4	5	6
		

SIGNAL STRENGTH/QUALITY MEASUREMENTS

In the LTE network, a UE measured two parameters for the reference signal to determine the strength and the quality of the signal: RSRP (Reference Signal Received Power) and RSRQ (Reference Signal Received Quality). 5G NR uses SS-RSRP (Synchronization Signal Reference Signal Received Power) and SS-RSRQ (Synchronization Signal Reference Signal Received Quality) to define signal level and quality.

As a rough guide, Table 10 shows what values were considered good and bad for the LTE/5G signal strength/quality values.

TABLE 10. LTE/5G SIGNAL STRENGTH/QUALITY VALUES

	RSRP (Signal Strength)	RSRQ (Signal Quality)
Excellent	> -85 dBm	> -10 dBm
Good	>-85 dBm to -100 dB	>-10 dBm to -15 dB
Fair	>-101 dBm to -120 dB	>-16 dBm to -20 dB
Poor	<-120 dBm	<-20 dBm

During examination of the data, we found the following factors were highly relevant on how LTE and NR performance compared:

- The LTE and NR were operating on different frequencies, and there were differences in link setup between LTE and NR respectively (e.g., transmit power, antenna gains, etc.). In addition, the configured system did not use carrier-grade antenna, so the reliability was not guaranteed.
- Different vendors would set up different mechanisms to transition between NR and LTE. For example, by design, the configured system would connect to NR only when the signal is great.

The results with data collected from March 4 will be discussed in the following sections.

RSRP Distribution (Test Results from March 4)

Table 11 shows the RSRP values along with the actual flight patterns from March 4. We observed that the UE received good to excellent signal strength at altitude of 10 ft.

RSRP Distribution (Test Results from March 4)

Table 11 shows the RSRP values along with the actual flight patterns from March 4.

We observed that the UE received good to excellent signal strength at altitude of 10 ft.

TABLE 11. RSRP VALUES WITH ACTUAL FLIGHT PATTERNS—MARCH 4

1	2	3
		
4	**5**	**6**
		

Table 12 covers the histograms of RSRP from March 4. UE was not able to connect in 5G NSA mode on this day. We observed that ranges of RSRP and RSRQ values were tighter compared to March 3 due to more consistent flight altitude. In addition, UE received higher signal strength compared to March 3

TABLE 12. HISTOGRAMS OF RSRP—MARCH 4

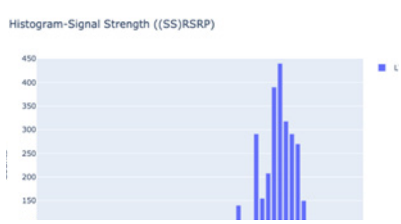
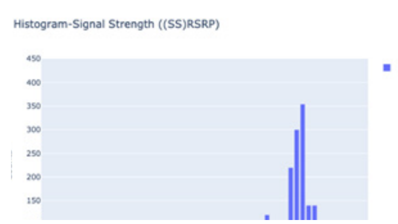
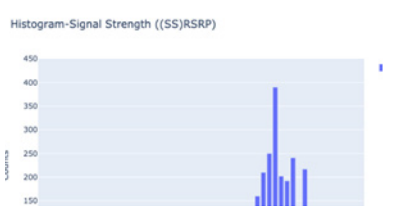
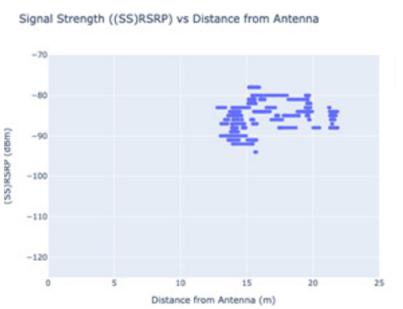
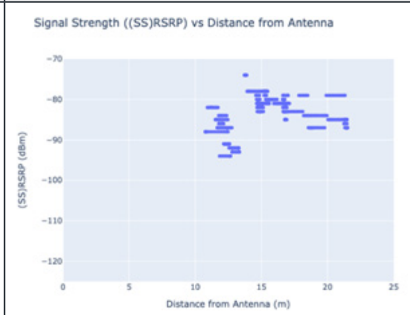
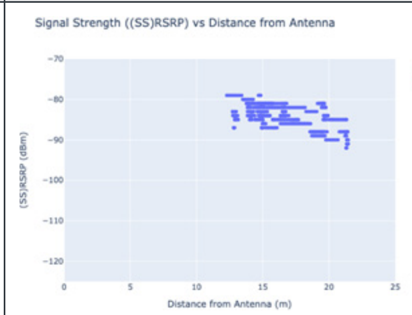
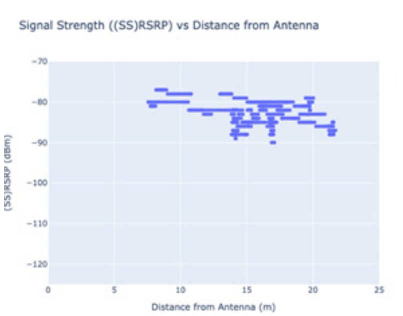
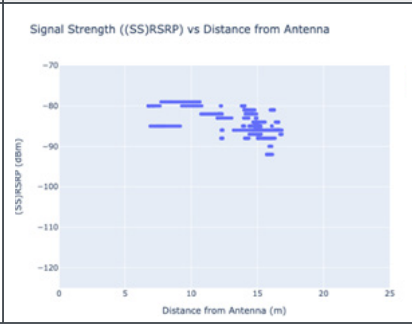
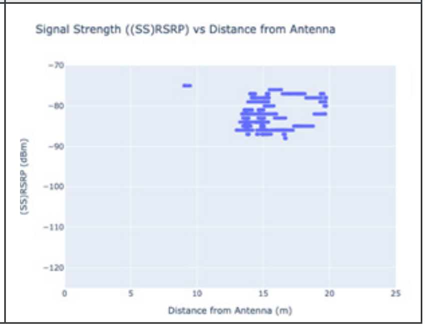
1	2	3
		
4	**5**	**6**
		

Table 13 shows the relation between LTE signal strength and the UE distance to BS antenna. The results show the UE received better signal quality when closer to the antenna and the ranges of RSRP values were tighter compared to March 3 due to the more consistent flight altitude.

TABLE 13. LTE SIGNAL STRENGTH AND THE UE DISTANCE TO BS ANTENNA—MARCH 3

<table>
<tr><th>1</th><th>2</th><th>3</th></tr>
<tr>
<td>
</td>
<td>
</td>
<td>
</td>
</tr>
<tr><th>4</th><th>5</th><th>6</th></tr>
<tr>
<td>
</td>
<td>
</td>
<td>
</td>
</tr>
</table>

RSRQ Distribution (Test Results from March 4)

Table 14 shows the RSRQ values along with the actual flight patterns from March 4. We observed that the UE received good signal quality at altitude of three meters most of the time. (One interesting observation was the UE got lower signal quality when the drone was in hovering mode—further investigation on the relationship between signal quality and drone speed is needed.)

TABLE 14. RSRQ VALUES WITH ACTUAL FLIGHT PATTERNS—MARCH 4

1	2	3
		
4	5	6
		

Table 15 covers the histograms of RSRQ from March 4. UE was not able to connect in 5G NSA mode on this day. We observed that ranges of RSRQ values were tighter compared to March 3 due to more consistent flight altitude.

TABLE 15. HISTOGRAMS OF RSRQ—MARCH 4

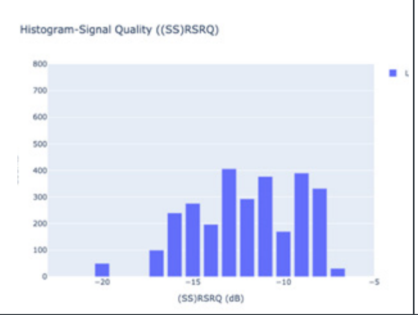
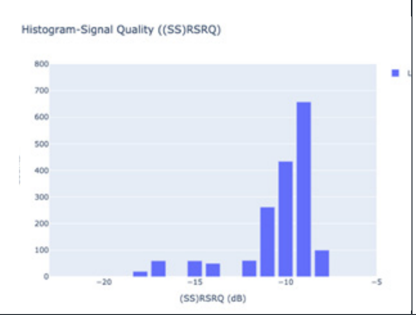
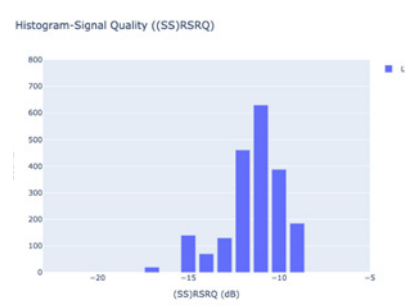
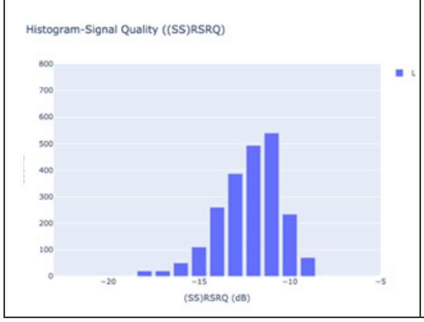
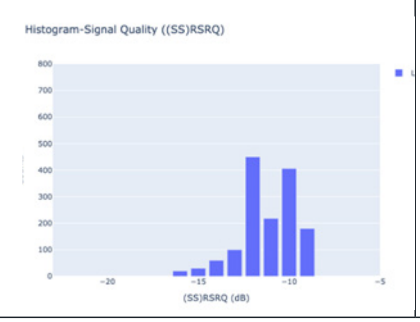
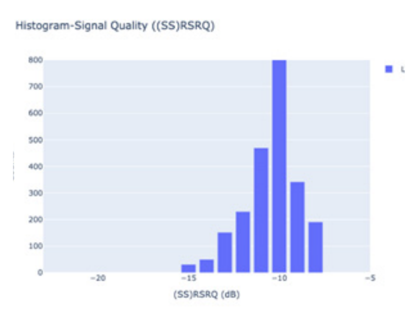
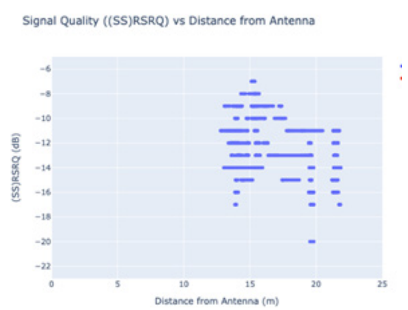
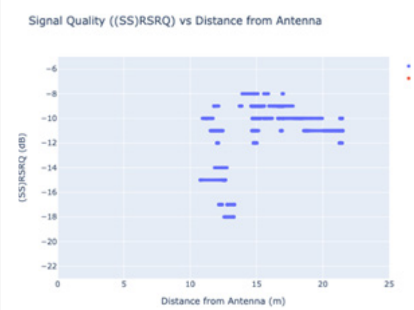
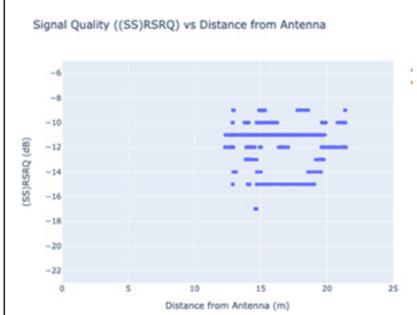
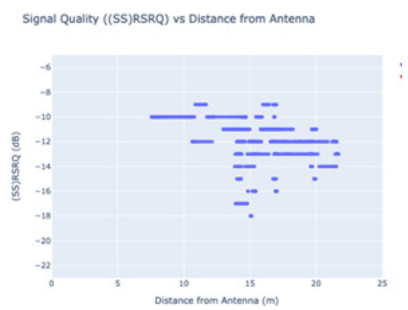
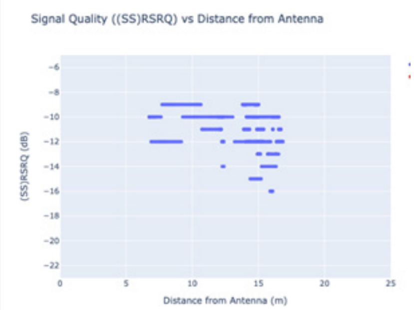
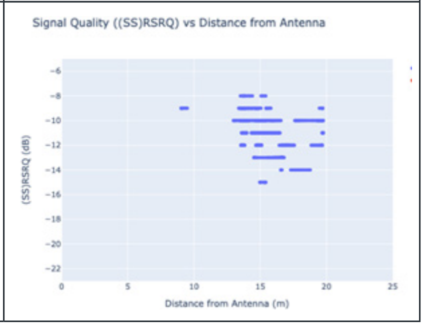
1	2	3
		
4	5	6
		

FIGURE 27. HISTOGRAMS OF RSRQ FROM MARCH 4

Table 16 shows the relation between LTE signal quality and the UE distance to BS antenna. In general, we observed that UE received lower signal strength and quality when further away from the antenna.

TABLE 16. LTE SIGNAL STRENGTH AND THE UE DISTANCE TO BS ANTENNA—MARCH 4

1	2	3
		
4	5	6
		

Conclusions and Insights

This report described the series of 5G UAS experiments performed by Virginia Tech (CCI/MAAP team) and MITRE Engenuity from March 2 to 5, 2021, at the VT Drone Park in Blacksburg, VA.

A 5G NSA network was established and connectivity with user equipment was established both in LTE and 5G NR. An UA flew using the 5G NR and LTE for both telemetry and commands. Data was gathered on the flight controller computer, the UE, Amari Callbox, and the GCS.

The experiments successfully demonstrated the use of 5G NSA networks for UAS command, telemetry, and payload and provided insights for future experiments.

The following was accomplished during the experiments.

- Integrated drone with 5G user equipment (UE)
- Implemented end-to-end communications between ground control station (GCS) and drone connected to a 5G NSA base station, including:
  - Drone C2 telemetry (sensor data from drone to GCS)
  - Drone C2 commands (commands from GCS to drone)
  - Real-time, in-flight video streamed from drone, to ground server
- Performed flight tests with the 5G drone operating in the following modes:
  - Loiter (held position and altitude based on GPS & barometric altimeter)
  - Auto (flight along a predefined path)
  - Guided (sent commands to proceed to each waypoint in real time, over cellular connection)

There were also insights achieved during these experiments that we plan to consider towards future experiments:

- Separate network and experiment validation activities: in this experiment, we took the challenge to deploy the cellular network and perform flight tests and connectivity experiments on the same day. Separating the process of validating the network operation and coverage, the end-to-end drone connectivity, and then the flight tests, helps greatly with the process of debugging and refining the solutions.
- Real-world field experimentation always incurs unexpected issues with hardware and with integration of parts. Experimentation risk must be anticipated and mitigated as much as possible (through inclusion of spare parts to the field, additional schedule margins, diagnosis tools).
- Know what to expect: modeling and simulation to reproduce the planned setup greatly help with understanding expected network performance ahead of the actual field experiments. The use of simulation and emulation environments can resolve issues faster and easier than trying to solve them in the field the day of the experiment.
- Rehearse: Dry runs of the experiment are very important to validate test procedures and operational issues. The software tools used for data- gathering and post-processing should be fully developed, tested, and validated prior to the experiment.

- Plan for bad weather: accommodate margins on schedule for weather (especially if experimenting in the winter).

Additional efforts are being planned beyond Phase 1B for experimentation, as part of the Open Generation activities. From what we learn from these experiments, we plan to publish data on 5G as a reliable method for C2 and DAA in BVLOS, and data on 5G performance for mission and payload. We also plan to report experiments and results and propose improvement methods to standards bodies.

We will create testbeds that will serve as 'experimental sandboxes' for the community. We expect this first vertical, UAS, to be the initial step towards our mission to advance 5G innovation and competitiveness in the U.S. and democratic societies, by advancing standards that support UAS operations with 5G, advancing data to support regulations for BVLOS operations, and advancing the business case for the several industry sectors (telecommunications, UAS, service providers).

Glossary of Terms

Acronyms	Definitions
3GPP	3rd Generation Partnership Project
5GC	5G Core Network
NR	New Radio
AGL	Above Ground Level
APN	Access Point Name
BS	Base Station
BVLOS	Beyond Visual Line of Sight
C2	Command and Control
CCI	Commonwealth Cyber Initiative
CIL	Cell Info Logger
CQI	Channel Quality Indicator
DAA	Detect and Avoid
DL	Downlink
DUT/SUT	Device and System Under Test
EPC	Enhanced Mobile Broadband
eMBB	Evolved Packet Core
eNB	eNodeB
EPC	Evolved Packet Core
FDD	Frequency Division Duplex
GCS	Ground Control System
gNB	gNodeB
GPRS	General Packet Radio Service
GTP	GPRS Tunneling Protocol
IP	Internet Protocol
KPI	Key Performance Indicator
LTE	Long Term Evolution
MAAP	Mid-Atlantic Avation Partnership
MAVLink	Multiple Input, Multiple Output
MIMO	Micro Air Vehicle Link

MME	Mobile Management Entity
mMTC	Massive Machine Type Communications
NAVIO2	Navigation Input/Output
NSA	Non-Stand-Alone
PCRF	Policy and Charging Rules Function
PDN	Packet Data Network
PER	Packet Error Rate
PGW	PDN Gateway
PRACH	Physical Random-Access Channel
QCI	QoS Class Identifier
QoS	Quality of Service
RAN	Radio Access Network
RF	Radio Frequency
RPi	Raspberry Pi
RSRP	Reference Signal Received Power
RSRQ	Reference Signal Received Quality
SGW	Serving Gateway
SNR	Signal to Noise Ratio
SS	Synchronization Signal
TCP	Transmission Control Protocol
TDD	Time Division Duplex
UA	Uncrewed Aircraft (also known as Unmanned Aircraft)
UAS	Uncrewed Aircraft Systems
UDP	User Datagram Protocol
UE	User Equipment
UL	Uplink
URLLC	Ultra-Reliable and Low Latency Communications
VLC	VideoLAN Client
VT	Virginia Tech

Footnotes

1. MITRE Engenuity Open Generation Position Paper: "Enabling 5G Innovation Leadership through Use Case-Driven Collaboration," October 2021
2. For more information see: additional Open Generation activities beyond the scope of this report.
3. FAA NextGen UAS Concept of Operations v2.0
4. 3GPP TR 36.777: Enhanced LTE support for aerial vehicles technical report
5. 3GPP TS 36.331: Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification technical specification
6. 3GPP TS 22.125: Unmanned Aerial System (UAS) support in 3GPP technical specification
7. 3GPP TS 22.261: Service requirements for the 5G system
8. 3GPP UAS UAV website
9. This data was not collected during the experiments
10. Linux operating system used is available at: Emlid Raspbian Image
11. More information is available from the VideoLAN Organization website.
12. Technical details for various Callbox product options are available from Amarisoft.
13. Technical details can also be found on the Amari Callbox Pro Datasheet.
14. Detailed technical specifications are available on the Poynting OMNI-600 web site.
15. Android Developers Reference: CellSignalStrengthNR Class.
16. Android Developers Reference: Location Class.
17. Android Developers Reference: CellInfoLte Class.
18. Android Developers Reference: Telephony ServiceState Class
19. Android Developers Reference: Telephony NetworkRegistrationInfo Class

List of Figures

List of Tables

As MITRE's tech foundation for public good, MITRE Engenuity collaborates with the private sector on challenges that demand public interest solutions, to include cybersecurity, infrastructure resilience, healthcare effectiveness, microelectronics, quantum sensing, and next generation communications.

For more information on the Open Generation consortium mission and progress of activities, please check our web page at: https://opengeneration.mitre-engenuity.org/ or contact lribeiro@mitre-engenuity.org

MITRE ENGENUITY™ | Open Generation™