IEEE Access
Multidisciplinary ⁞ Rapid Review ⁞ Open Access Journal

Received April 12, 2019, accepted May 9, 2019, date of publication May 20, 2019, date of current version June 6, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2917851

A Safe, Open Source, 4G Connected Self-Flying Plane With 1 Hour Flight Time and All Up Weight (AUW) < 300 g: Towards a New Class of Internet Enabled UAVs

PETER J. BURKE, (Senior Member, IEEE)
EECS, BME, Chemical and Biomolecular Engineering, Materials Science and Engineering, and Chemical and Materials Physics, University of California at Irvine, Irvine, CA 92697, USA
e-mail: pburke@uci.edu

Funding for this work was provided by the author's private finances. The design, construction, assembly, bench testing, and coding work was performed in the author's garage.

ABSTRACT Here, we present an open-source cloud (4G) connected and controlled self-flying airplane based on a secure, encrypted flight stack with all up weight of <300 g as an example of a new class of unmanned aerial vehicles (UAVs). The avionics portion (including the flight controller, 4G cellular data modem, and on-board companion Linux computer) weighs only 40 g, which is about 10 times lighter than previous avionics package achieving the same mission: 4G secure encrypted robust self-healing Internet connectivity for video and telemetry, autonomous flight capability without the need for a pilot in local proximity to the UAV, and GPS positioning. With a flight time of 1 h, this represents the lightest, longest flight time 4G UAV ever demonstrated. Enabled by the convergence of advances in multiple hardware and software systems and disciplines, this effectively defines a new class of the UAVs, distinct from larger multirotors for aerial surveys and photography and smaller nano-quadcopters with short flight times and ranges, defined by the characteristics of inherently stable aerodynamic flight in case of hardware failure (such as motor failure), inherently long flight times, inherently long range due to the efficiency of a wing over quadrotor, and Internet connectivity for control from any point on earth to virtually any point on earth.

INDEX TERMS UAV, cloud computing, Internet of Things.

I. INTRODUCTION

Howard Hughes and his engineers remotely piloted a scale model of the largest wingspan plane ever built, the Spruce Goose [1]. 50 years later, the US Military regularly fields and deploys UAVs weighing over 10 tons with pilots located in a different part of the world [2]: Clearly, the use of long range telemetry for unmanned aerial vehicle (UAV) guidance and control is well established in the military for full sized UAVs weighing up to 100 kg (e.g. Predator, Reaper, etc.) However, in neither the civilian nor the military sector has use of the internet for long-range control of airframes in the micro (sub 500 g) and nano (sub 250 g)- class been demonstrated.

Here, similar to the US military 10 ton drones, but much smaller in size than Howard Hughes would ever have dreamed, we present detailed plans and demonstrations of an example of a new class of internet connected micro-UAVs (Figure 1, Figure 2, Figure 3) which can be controlled via cellular telephone networks from virtually anywhere on the planet to virtually anywhere on the planet.

This class of systems is enabled due to multiple recent advances: 1) Recently (spring, 2018) the Ardupilot code base [3], [4], arguably the largest open-source autopilot software project in the world installed on over one million vehicles [5], was ported [6] to Chibios, a micro-operating system [7], [8]. This enabled abstraction of the high level software from the hardware details. In particular, it enabled porting of the code to much smaller and less expensive STM 32 bit microcontrollers originally developed for small racing drones [9]. Instead of a 50 g flight controller, a 5 g controller can do the job at on the order of ~\$10/board rather than ~\$100/board, see below. (To boot, it provided an on-screen-display HUD (heads up display) type overlay and current sensor as part of the board, which were previously

The associate editor coordinating the review of this manuscript and approving it for publication was Qingchun Chen.

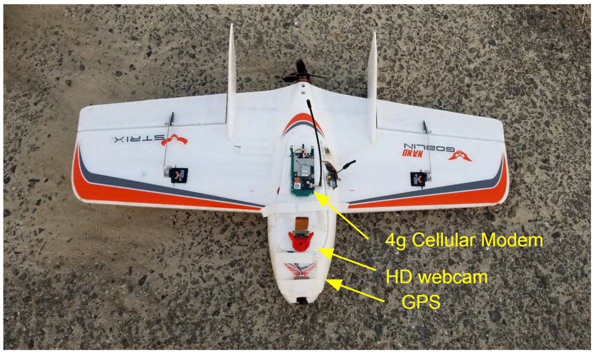


FIGURE 1. Picture of first generation internet connected UAV. Once this prototype successfully demonstrated proof of concept, the external components (GPS, 4G modem, camera, 5.8 GHz video transmitter) were incorporated inside the fuselage.

two additional boards.) 2) The development of the Raspberry Pi Zero W [10] enabled a full Linux board weighing only 9 g at $10, much lighter than the 50 g or more of prior on-board computers [11]. 3) 4G modem chipsets made for cellular data are available at increasingly low prices, weighing in at 10-20 g and available for under $100. 4) The availability of full stack Linux machines in the cloud through various service providers for literally pennies (a few $ per month) such as Amazon Web Services (AWS) [12].

This new class of aerial vehicles, of which an example is presented in full detail herein, is not enabled by any specific breakthrough of a single component or sub-system, but rather the convergence of maturity of multiple engineering disciplines, including advances in these key technical fields: Linux single board computers, cloud computing, software abstraction, secure encryption, and ubiquitous wireless communications. Taken collectively, these have enabled for the first time the ability to perform sophisticated, remote flight at very low cost with long flight times.

In this example work, all the code is open source and uses the latest encryption technologies for security. Furthermore, the low weight of the airframe dramatically increases safety (see below): a foam based airframe of under 250 g is much less likely to cause damage or injury in comparison to a metal based airframe that weighs 10 times as much.

Furthermore, because of the inherent increased safety factor, these systems can be designed and prototyped rapidly, and used as ''testbeds'' to help with lessons for larger brethren. This light weight micro avionics system is made from commercially available parts for under $100 (see below). Although compatible with virtually any airframe type, we chose a commercially available flying wing design [13] due to its low weight and simplicity to demonstrate the technology, as well as additional safety factor of a pusher prop configuration, which minimizes any risk of personal injury in case of an accident.

This case exemplifies a new class of UAVs, and can be built by anyone using the open-source hardware and software described herein. By comparing this to other genres of UAVs, this paper provides a broad overview of the field, with detailed side-by-side comparisons to state of the art in the other more mature classes of UAVs.

To summarize of the contributions of this work to the field, it demonstrates the lightest, longest flight time 4G connected UAV ever made, which uses a new self-healing internet architecture developed in this work, together with extensive failsafe contributions, defining essentially a new, safe, secure, and open-source class of UAVs. This is a unique self-healing architecture, not previously described in any literature. These contributions are put into context through a review of available avionics platforms and UAV systems.

II. REVIEW OF FLIGHT CONTROLLER HARDWARE & SOFTWARE OPTIONS

We review the software and hardware options for flight controllers, summarized in Table 1.

A. FLIGHT CONTROLLER HARDWARE: BOARD

The flight controller is the ''brain'' of the system, and consists of a microcontroller (typically 32 bit STM based) with custom firmware (described below). The micro-controller takes as an input the craft orientation, position, and other sensors, as well as pilot input through a radio or internet link, and computes outputs including motor power, control surface deflection (e.g. rudder, elevator, aileron). Various control modes are available with increasing levels of automation. The simplest mode, pass through, sends pilot commands directly to control surfaces and motor power for complete manual control. The most automated controls can auto-takeoff, fly waypoint missions, and auto-land, with zero pilot input. (For an example, see ''self-flying plane'' below).

Typically, a barometer for altitude, an accelerometer for orientation and acceleration are included on the custom board as sensors at a minimum, and multiple servo outputs for control. In some cases, an on-board current sensor monitors total power consumption of the system. Additional digital interfaces such as UART and I2C can enable off-board sensors such as GPS, compass, pitot tube airspeed sensor, LIDAR, and sonar.

As the video signal from an on-board camera is usually downlinked as an analog signal, recent flight controllers have begun to integrate an configurable on-screen-display (OSD) to give the pilot quantitative information in a customizable format, such as airspeed, altitude, GPS orientation, an artificial horizon, etc. This functions as a heads-up display (HUD) and can enable manual or auto flight control even at night or in clouds, for the UAV version of Instrument Flight Rules (IFR).

B. FLIGHT CONTROLLER SOFTWARE: FIRMWARE & OS

The firmware is typically compiled from custom code for ''bare metal'' performance. Some firmware can run on top of simple operating systems. The most used open source firmware packages are Ardupilot and Cleanflight/Betaflight/iNav, originally developed for fixed wing and multirotor platform, although both now can function to control a large variety of vehicles. All of the listed software

FIGURE 2. Historical development of UAVs: A) Howard Hughes and his engineers pilot a scale model of the now famous Spruce Goose, the largest wingspan airplane ever built. B) The US Military consistently deploys remotely piloted UAVs weighing 10 tons or more. C) (Author on laptop and his plane above him.) With just a laptop and a few hundred dollars of components, a much safer, quieter, and discrete UAV can be deployed securely using open-source software and fully encrypted internet communication protocols as demonstrated in this paper.

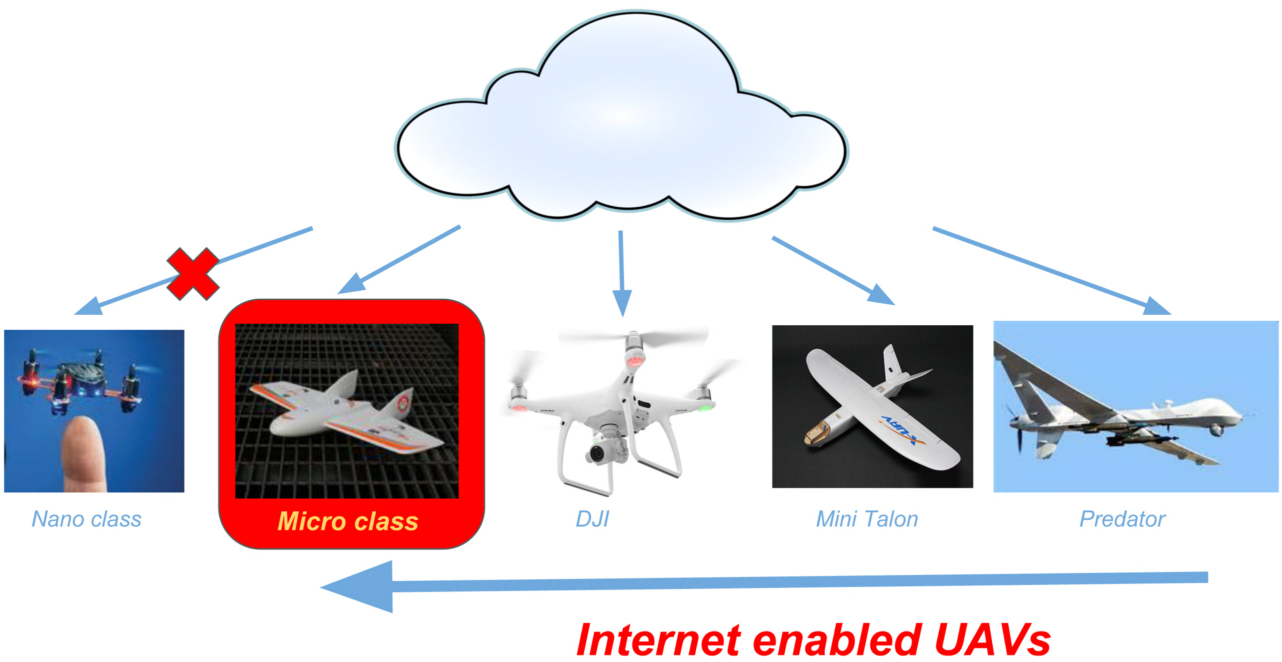


FIGURE 3. Towards a new class of micro-UAVs: Traditional internet enabled UAVs weigh over 2 kg; this new class is smaller (under 300g), quieter, and safer [2], [13]–[15].

options function well and enable GPS guided or manual flight successfully. Some are more optimized than others for specific mission profiles (e.g. close in racing, long range, etc.) Recently [28] (spring 2018) the Ardupilot code base was ported to Chibios OS allowing hardware abstraction and operation on a variety of different custom boards. While there

TABLE 1. Representative selection of commercially available GPS guided flight controllers [16]–[27].

Flight Controller	CPU	Year	Wt. (g)	Dim. (mm)	Volume (mm^3)	Cost ($)	OSD+I $	OSD+I Wt (g)	Total weight (g)	Total cost ($)	Software
Pixhawk 2	STM32F427	2013	38	81x50x15	60750	200	28	21	59	228	Ardupilot
Pixhawk 4	STM32F765	2018	16	44x84x12	44352	211	28	21	37	239	Ardupilot
EagleTree Vector	Proprietary	2015	21	65x33x14	30030	238	-	15	36	238	Vector
EagleTree MicroVector	Proprietary	2016	13	38x38x10	14440	100	**	**	13	100	Vector
Arkbird	Proprietary	2013	146	120x90x70	756000	190	-	-	146	146	Arkbird
Omnibus F3 Femto	STM32 F3	2016	2	20x20x3	1200	23	**	**	2	23	Ardupilot/iNav/Betaflight
Omnibus F3 Nano	STM32 F3	2016	3	25x25x3	1875	26	**	**	3	26	Ardupilot/iNav/Betaflight
Omnibus F4 Nano	STM32 F4	2017	5.4	25x25x3	1875	30	**	**	5.4	30	Ardupilot/iNav/Betaflight
Omnibus F4 Pro	STM32 F4	2017	8.4	30x30x3	2700	35	-	-	8.4	35	Ardupilot/iNav/Betaflight
Omnibus F7 Pro	STM32 F7	2018	7	30x30x3	2700	60	**	**	7	60	Ardupilot/iNav/Betaflight

Current sensor: Hollybro PM02; wt. 20 g; cost $15; OSD: MinimOSD, wt. 1 g; cost $13; Eagletree, Arkbird cost/wt. includes GPS; - means integrated on flight controller board; ** Isense not avail; I=current measure

TABLE 2. Craft weight breakdown by avionics, battery, and airframe.

Item	Weight (g)
Avionics	40
Battery (2S 18650 Li Ion)	104
Airframe/motor/ESC/servos	156
Total	**300**

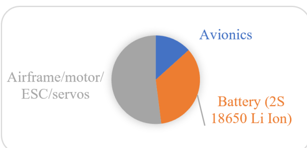
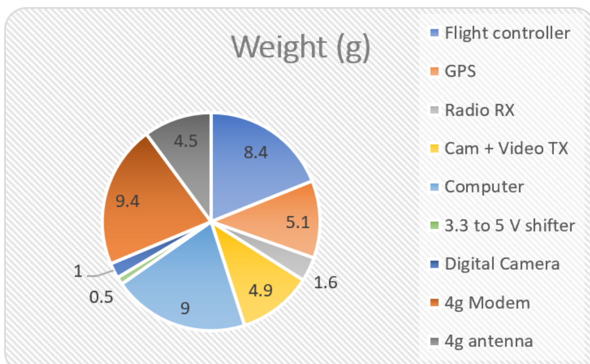


TABLE 3. List of avionics components. (Cost as purchased 2018; prices may have changed; weight measured in author's garage workshop.).

Component	Model #	Cost ($)	Weight (g)
Flight controller	Omnibus F4 Pro	26	8.4
GPS	BN-220	11	5.1
Radio RX	TBS Nano	30	1.6
Cam + Video TX	TX05	9	4.9
Computer	Rasp. Pi 0 W	10	9
3.3 to 5 V shifter	Cylewet	1	0.5
Digital Camera	Mini-Size Cam.	15	1
4g Modem	USB720L	80	9.4
4g antenna	TS9	1.5	4.5
Total		182	39.9

Alternative:

VTX	TX03	9	2.7
HD Camera w/SD	Caddyx	65	12



are a variety of very capable autopilot hardware and software solutions available for both fixed wing and multi-rotor platforms, the porting or Ardupilot to Chibios enabled some uniquely powerful features of Ardupilot on very small boards (Table 1), specifically the ability to communicate with an on-board companion computer in a seamless, open source way, for the first time.

III. HARDWARE STACK (AVIONICS)

We now describe the particular implementation chosen for our example design. The hardware stack is an Ardupilot based stack. The all-up-weight (AUW) is under 300 g, as shown in Table 2. The complete avionics list is given in Table 3. The schematic of the avionics is in Figure 4.

A. FLIGHT CONTROLLER

The flight controller (Omnibus F4 Pro [22]) is based on an STM32 bit F4 processor. An integrated on board current sensor (50 mΩ resistor) provides power management and an integrated on board video overlay (Maxim Integrated MAX7456 single-channel monochrome on-screen display (OSD) or similar) enables a custom heads up display (HUD) for the analog video feed to the ground control station.

B. POWER

A 2S 18650 Li Ion 3200 mAh battery pack is used for power. The electronic speed controller (ESC) provides a 5V 2A DC voltage through an integrated battery eliminator

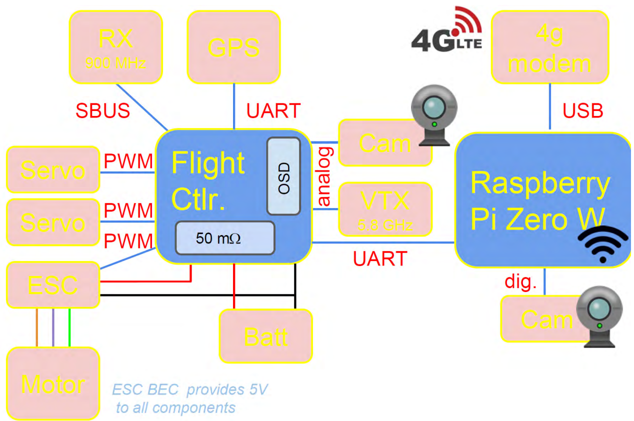
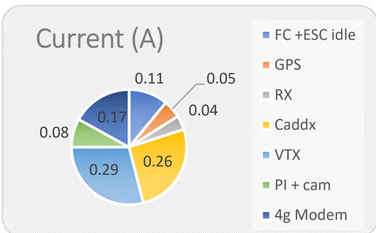


FIGURE 4. Schematic of avionics.

TABLE 4. Avionics power budget (measured in authors garage workshop).

Item	Current (A)
FC +ESC idle	0.11
GPS	0.05
RX	0.04
Caddx	0.26
VTX	0.29
PI + cam	0.08
4g Modem	0.17
Total (A)	**1**



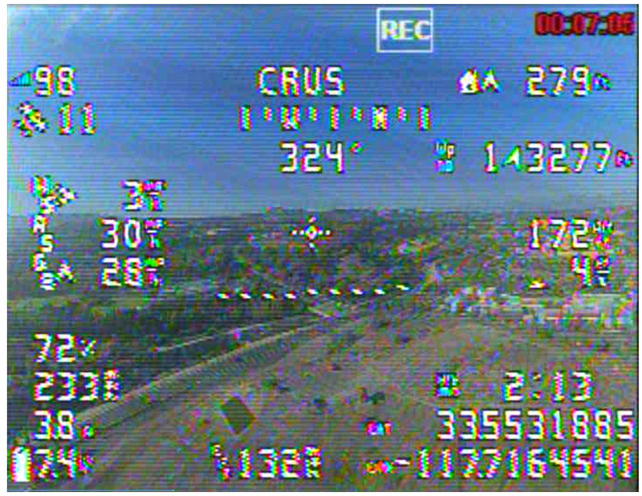


FIGURE 5. HUD. This can be customized to include, e.g, airspeed, windspeed, groundspeed, altitude, GPS coordinates, sink rate, battery voltage and current, # of GPS satellites, RSSI for radio control, flight mode, heading distance to home, etc.

circuit (BEC), which is sufficient for the servos, flight controller, GPS, onboard computer, 4G modem, and both on board cameras. (The power budget of the avionics is shown in Table 4) The current draw with throttle off is 1 A. At cruise, the total current draw is around 3 A, giving an estimated 1 hour flight time (see endurance, below).

C. RADIO RECEIVER

A 900 MHz radio receiver (TBS Crossfire Nano [29]) with linearly polarized dipole antenna is used, communicating with the flight controller via SBUS protocol.

D. GPS

A BN-220 NEO-M8N GPS GLONASS Antenna Module Ublox was used for GPS with a UART connection to the flight controller. Typically, over 15 satellites are locked at any given time, with an HDOP of <1 as a precondition to arm.

E. ANALOG VIDEO

An 800 TVL analog camera with HD on 1080p recorder to an SD card is used (Caddyx Turtle V2), together with a switchable 5.8 GHz transmitter with up to 200 mW of transmit power (Eachine VTX03), together with a linearly polarized antenna with 2 dB gain.

F. COMPANION COMPUTER

In order to minimize weight, a Raspberry Pi Zero W [10] is used. Power to the Pi is from the 5 V rail, sourced from the BEC. A 3.3 V to 5 V logic level shifter is needed between the flight controller UART and the Pi Zero UART, as the Pi uses 3.3 V. The Pi will be damaged if this is not used.

G. DIGITAL VIDEO

A 1080 P (SainSmart Camera Module 5MP Mini Size for Raspberry Pi Zero) Raspberry Pi camera with 15-pin MIPI Camera Serial Interface is used, managed by the Pi. This is also made available as a webcam through the internet while the craft is in the air (see below).

H. 4G CELLULAR MODEM

A Verizon 4G USB modem is used (USB730L [30]). While older modems have trouble with Linux connection and drivers, this modem works out of the box. The modem provides an on board DHCP server. The stock modem weight is 45 g. We found that the antenna in the case, the case itself, and the bulky USB connector can be removed, and replaced with a linearly polarized external antenna (TS9). Removing the case provides a weight of 15 g (with TS9 antenna). Data rates of 10 Mb/s are readily achievable. A single use data only plan of 1 Gbyte/month is $15/month at Verizon as of this writing. This is plenty of bandwidth for the telemetry and video.

I. AVIONICS POWER BUDGET

The avionics power budget is about 1 A. See Table 4.

IV. SOFTWARE STACK (FLIGHT CONTROL)

A. FLIGHT CONTROL

Flight control is via Ardupilot 3.9.2. It runs on Chibios.

B. ON SCREEN DISPLAY (HUD)

The HUD is overlayed on the analog video. The software enables configuration of the screen. An example is shown in Figure 5.

FIGURE 6. **Ground control station. The Windows 10 PC runs Mission Planner to control the plane over the internet. The 7" monitor on top receives the analog video broadcast directly from the craft to the ground at 5.8 GHz.**

C. ON-BOARD COMPUTER

The Raspberry Pi Zero W runs Linux, the standard Raspbian distro. It handles the networking, as discussed below in the networking setup. It also handles the video and acts as the webcam server.

D. GROUND CONTROL STATION (GCS)

A Windows 10 PC running Mission Planner 1.3.62 acts as the ground control station. In addition, an analog monitor with a low noise 5.8 GHz diversity receiver (Aomway HD518 with 14 dB gain patch antenna (Aomway ANT05) and 3 dB gain rubber duck antenna) with 7'' monitor displays and records the analog video link. The GCS is shown in Figure 6.

V. NETWORKING STACK

The 4G modem provides internet connectivity for the UAV. All communications are encrypted via SSH (Secure Shell) [31]. A cloud based Linux server is used as a fixed point to orchestrate the connection. Figure 7 shows the networking stack overview. The overall architecture is designed to robustly and elegantly respond to lost links at various points in the network, i.e. the network connection is encompasses a ''*self-healing*'' design at multiple software and hardware levels, some built into the design of the internet itself, others through UAV specific code and settings.

The UAV on-board companion computer (Raspberry Pi Zero W) is connected to a cloud Linux server (an AWS instance running Ubuntu 18) via a reverse SSH tunnel [31]. The ground station computer (Windows 10) is connected to the same Linux cloud station through a reverse SSH tunnel. AWS is used as the service provider, but any Linux computer with 24/7 internet access with a fixed IP address would work. The advantage of this configuration is the low cost (few dollars/month of AWS fees) of the cloud computer, together with the flexibility to have both the GCS and UAV not requiring fixed IP addresses (which is impractical and expensive), and able to function securely even behind firewalls. Since all communications over the internet are encrypted, the system is secure.

The entire codebase and is available through a GNU license at gitlab.com/pjbca/4Guav, as well as detailed configure scripts for each Linux computer, a wiki, and a field manual.

A. UAV-CLOUD CONNECTION

On boot, a script in the Raspberry Pi initiates the reverse SSH tunnel to the AWS instance. If it is disconnected for any reason, the script attempts to reestablish the connection via the auto-SSH package.

B. GCS-CLOUD CONNECTION

As part of the standard operating procedure, the user must manually enable the GCS-AWS instance reverse SSH tunnel. A set of instructions is provided to monitor the SSH tunnel status via a terminal. Briefly, in a terminal to the AWS station, the user invokes the stcptrack command (a Linux package), which lists all active SSH connections into and out of the AWS server. This allow the user to monitor all links in the connection in real time. Thus, the user will be able to determine immediately if the SSH connection is broken, and take appropriate action (see failsafe/lost link section below). Once all the SSH tunnels are established, the mission planner software can be connected to the UAV.

C. TELEMETRY PROTOCOL

The MAVLink protocol [32] is used to pass telemetry and commands to/from the GCS to the UAV FC. A lightweight open-source set of forwarding utilities maintained by Intel (MAVLink Router [33]) are used to traffic the MAVLink

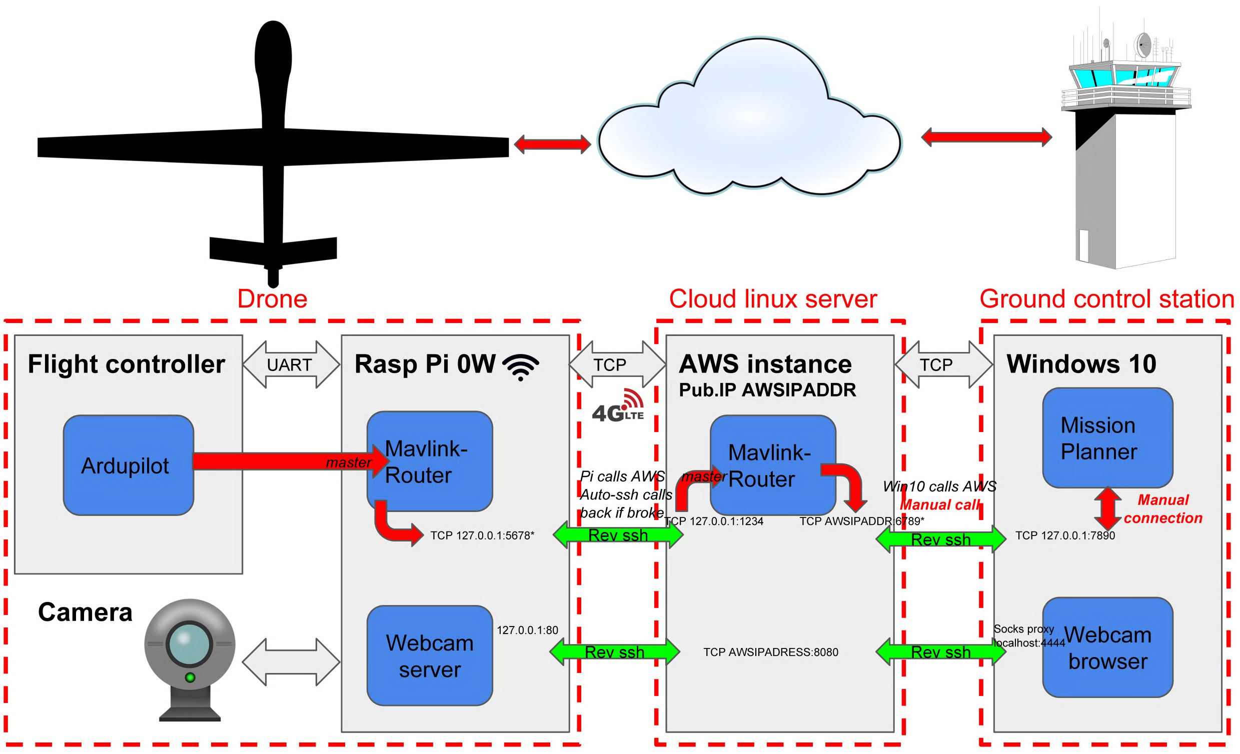


FIGURE 7. **Networking configuration.**

packets back/forth through the various links: The Raspberry Pi and the AWS instance each run MAVLink Router.

D. VIDEO CONNECTION

The video connection is similar to the telemetry/control connection above: For the video, there is another reverse SSH from the Pi to the AWS instance, and from the GSC instance to the GCS. The Pi initiates a webserver which provides video data to a web browser on the GCS PC, using a socks 5 proxy SSH tunnel. Thus, the video is secure and encrypted for the open internet traffic. In addition, the Pi records the 1080p HD video to its on board USB memory card.

E. Wi-Fi

The Wi-Fi hardware of the Pi Zero is used to create a local hotspot so that the user can log into the Pi from the ground when it is within range of the Wi-Fi (typically 30 m).

VI. CONTROL PROTOCOL & FLIGHT MODES

Two independent, redundant method can be used to pilot the UAV: A standard 900 MHz radio control based on TBS Crossfire hardware, and the internet control (Figure 8).

A. FLIGHT MODES

Ardupilot defines over a dozen flight modes. The manual mode allows pilot input into a joystick to control the throttle and elevon positions manually for a pure "stick and rudder" flying experience. Auto mode executes waypoint missions

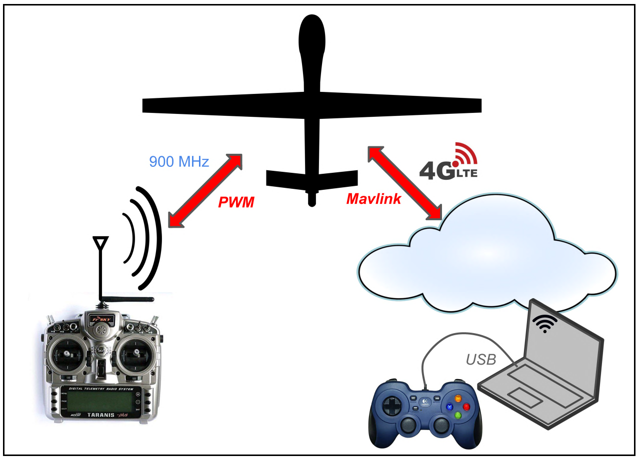


FIGURE 8. **Control protocols.**

and flies the craft autonomously to a fixed set of waypoints; landing and takeoff can each be defined as waypoints. A particularly important mode is return to launch (RTL) where the craft flies autonomously back to its takeoff position. This is useful for lost link failsafe protocols; see below.

Both the internet and radio control can be used to pilot the craft and switch between all these modes. The only difference between the two control links in terms of functionality is the ability to edit waypoint missions, which can only be done with internet control (not the radio control). However, waypoint missions are usually pre-defined prior to launch so, for most use cases, both the radio and internet control are equivalent and redundant.

B. RADIO CONTROL

Radio control sends joystick positions for throttle, pitch, roll, as well as flight mode (e.g. manual, return to launch, etc.). A programmable FrySky Taranis hand held radio running OpenTX firmware [34] with a TBS Crossfire plug in with up to 1 W of transmit power at 900 MHz is used. An 8 channel PWM protocol serves as the underlying information encoding in this link.

Note the TBS can also provide MAVLink telemetry packet forwarding back/forth over the radio link. That is not shown in this paper but can be a third method to network the craft to the GCS.

C. INTERNET CONTROL

The internet control is via MAVLink packets. The Mission Planner software [35] provides a GUI (Figure 10) which shows the craft position on a map and its orientation on a HUD with artificial horizons. Hundreds of raw and scaled parameters can be displayed in real time about craft performance through a customizable set of interfaces.

In general, there are two modes of internet control: with and without a USB joystick.

1) INTERNET CONTROL WITH USB JOYSTICK

A USB joystick/gamepad such as a Logitech Gamepad F310 (Figure 8) can be used to control the craft just as the radio control. Depending on the bandwidth, the lag is typically 0.5 seconds from command input to servo response. Therefore, although purely manual piloting is possible with mode, it is not recommended. Stabilized or cruise mode are more appropriate.

2) INTERNET CONTROL WITHOUT USB JOYSTICK

The user can click with a mouse at any point on the map and say ‘‘fly here’’. In addition, all the flight modes can be enable, e.g. RTL, execute waypoint mission, etc. This method is appropriate for auto type flight modes.

VII. PERFORMANCE

Multiple aspects of performance have been tested and are presented here. No unsolved problems remain.

A. TERRAIN FOLLOWING

Ardupilot provides a terrain following mode which maintains the craft at a fixed altitude above the ground. The internet is used to collect terrain data as the craft flies. This is useful for flights in mountainous terrain, and also for return to home failsafe condition to avoid flying into terrain during the return to home phase.

In Figure 9, we show an example of the craft loitering around home with a 50 m radius. The home point is on a steep hill, so as the craft loiters, it ascends/descends autonomously as it flies in the uphill/downhill direction, respectively.

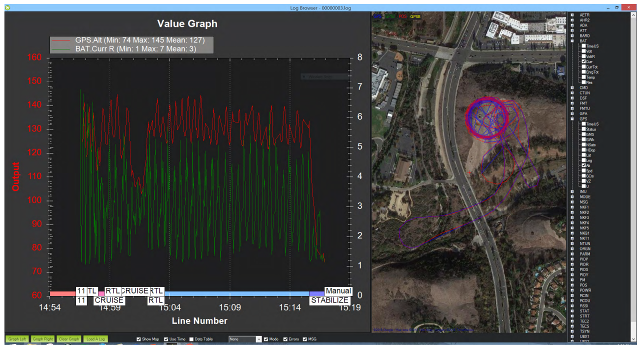


FIGURE 9. Terrain following. When circling home, MSL goes up and down to ensure AGL is constant. (Home point is on a steep sloping hill.) Also when flying in valley, MSL goes down to keep AGL constant in low valley (dip in MSL at 15:00 hours). Current high in climb; low in descent. Avg. current is 3 A. Altitude on left axis is in meters MSL. Right axis is current in A. Horizontal axis is local time (PST).

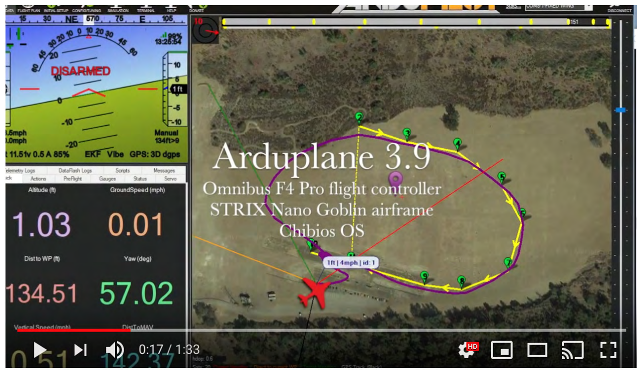


FIGURE 10. Autonomous flight path including auto-takeoff, waypoint mission, and auto-landing.

Monitoring the total battery current shows the throttle also oscillates as the craft ascends/descends, between 1-5 A. The throttle idle current is 1 A due to the avionics (see above), so the craft is actually in a controlled glide when heading downhill.

In order to avoid flying into a cliff, the software has a ‘‘look-ahead’’ feature, which automatically begins an ascent based on the pre-programmed maximum climb speed to avoid collision with the terrain [36]. The look-ahead distance is set by default to 2 km, but can be adjusted by the user.

B. FULLY AUTONOMOUS FLIGHT: SELF-FLYING PLANE

In order to demonstrate completely hands off, autonomous flight, we programmed a mission with the following features:

- Auto takeoff
- Waypoint mission
- Auto land

Figure 10 shows the pre-programmed flight path, as well as the actual flight path flown by the craft. The mission was monitored during flight in case the pilot needed to take manual control. This demonstrates a completely autonomous, self-flying airplane with no pilot input during the entire flight.

This was also demonstrated on another V-Tail airframe (ZOHD Nano Talon) and over 16 successful missions

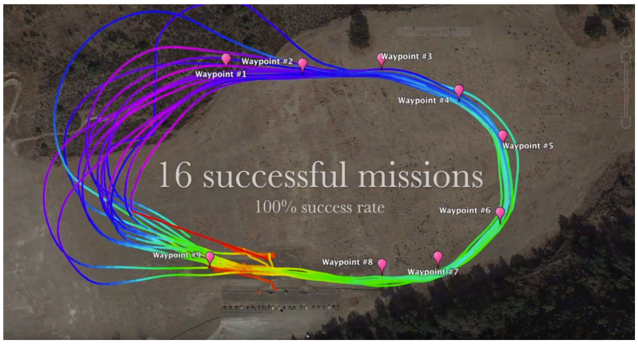


FIGURE 11. Google Earth generated flight path from telemetry logs of 16 separate autonomous flights based on the same waypoint mission, which included auto-takeoff and auto-land sequences. Color indicates relative altitude, with red ground level, and purple highest level reached during flight (appr. 100' AGL).

FIGURE 12. Approach and landing path of 16 separate flights shows Autoland feature to be consistent and reliable, and the landing position was consistent to within a few meters.

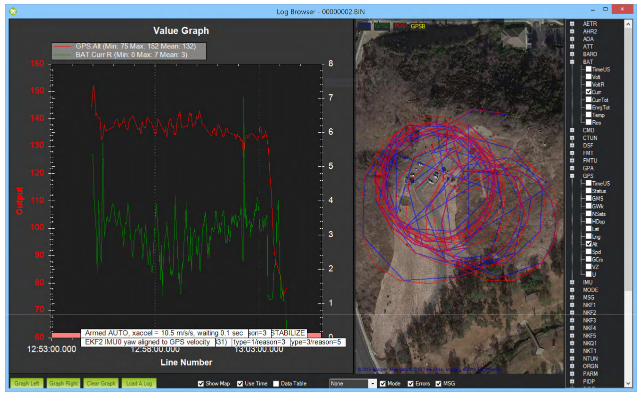


FIGURE 13. Loiter flight at constant altitude. Altitude on left axis is in meters MSL. Right axis is current in A. Horizontal axis is local time (PST). Average current is 3 A.

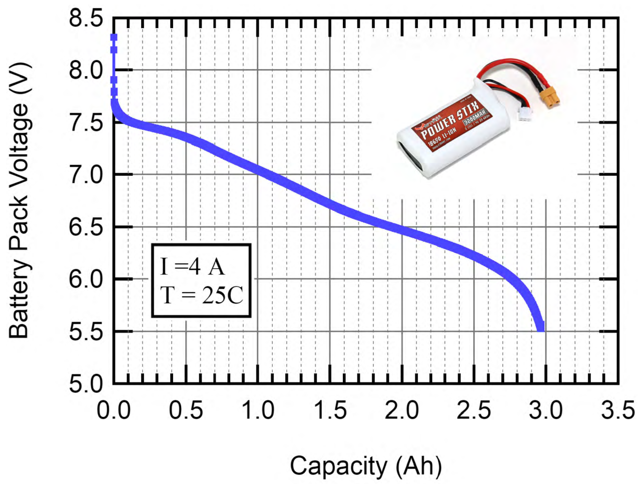


FIGURE 14. Li Ion battery pack discharge curve for two 18650 cells in series (2S). For 3 A cruise, a 1 hour flight time is possible with stock 3200 mAh batteries. Higher capacity batteries (up to 3500 mAh) are available but were not used in this setup.

based on the same preprogrammed missions were flown (Figure 11, Figure 12). Landings were smooth and successful, even in the presence of mild crosswind. Video from on-board and ground based cameras as well as detailed analysis is available on the authors YouTube channel (see Appendix for links).

C. ENDURANCE

In order to test the endurance, we performed a loiter flight at constant altitude to test the cruise throttle averaged over upwind/downwind flight directions. We found the cruise throttle for 30 mph flight around 3A (Figure 13).

We next tested the battery capacity of the stock 2S Li Ion battery pack provided by the airframe manufacturer (Strix), which is rated at 3200 mAh (Figure 14). The measurement setup was a MakerHawk UM25C USB power meter with a MakerHawk 150 W adjustable electronic load. The result clearly shows that the batteries can maintain a load of 3A with over 5.5 V on the pack for a capacity of 3 Ah, i.e. for a whole hour of flight time. The equation which relates the flight time T[hrs.] to the cruise current I_{cruise} [A] and the battery capacity C[mAh] is:

$$T[\mathrm{h}] = 1^{*}_{\mathrm{e}-3}\mathrm{C}[\mathrm{mAh}]/I_{\mathrm{cruise}}[\mathrm{A}] \quad (1)$$

Higher capacity packs of 3500 mAh are commercially available, and 3S packs can also be used, to further extend this flight time. Lower cruise throttle ratings can extend the flight times it even further.

D. WIND PERFORMANCE

This new UAV is so light that the readers may care about the stability in the wind. We flown the UAV in winds up to 15 mph without incident. (As a reference, the stall speed is 20 mph). The airframe has a very low cross section to the wind and performance is solid even in windy conditions. The flight stabilizer provides additional stabilization in the wind. The windspeed was estimated based on the windsock on the ground. The Ardupilot software also provides wind estimates based on the GPS speed vs. the required throttle to maintain that speed. In some cases, the ground speed can be as low as a few mph, but the craft still maintains stability and control in our experiments.

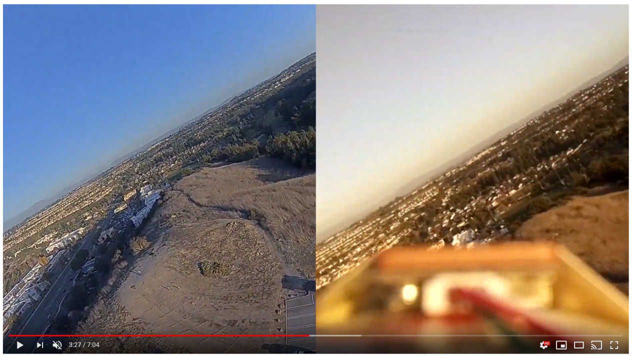


FIGURE 15. Webcam vs. HD camera video.

E. VIDEO

An example of the webcam video compared to an optimized high quality HD cam is shown in Figure 15. The webcam image is stored in HD format to an on-board SD card for later retrieval. The real-time webcam footage was smooth when over a local Wi-Fi connection to the onboard computer, but choppy over the internet connection due to the additional lag. In addition, the Raspberry Pi camera did not show sufficient dynamic range to view terrain features when looking directly into the sun. Perhaps in the future Pi compatible cameras will become available with dynamic range matching that of the existing FPV optimized cameras. However, the flight video was more than sufficient for navigation and situational awareness purposes. Flying using the webcam video in auto modes is more than enough quality to provide the pilot with an additional set of location and terrain information.

VIII. LOST LINK (FAILSAFE) BEHAVIOR

Of particular interest is the craft behavior on lost link. This is especially a concern, for example, when flying from one cell tower to the next: Although most modern protocols handle the cell-tower to cell-tower handoff seamlessly from the user point of view, there is always the chance the internet link will be temporarily lost and restored again. In addition, there is a chance that the craft will fly into a region where the service is not provided for 4G internet. Finally, since one use case considered is that of remote location control from anywhere on earth, the radio control link could be lost, or even absent from the get-go. Although the link distance for TBS Crossfire can be as large as 100 km [37], there may be cases where the radio control is located further than this, or the craft flies behind an obstruction (such as natural terrain or man-made obstructions such as tall buildings). Therefore, it is critical in this envisioned application space to have a sound fail-safe and lost-link behavior defined so that the operator can safely and reliably pilot the craft.

Below, we analyze each of the possible scenarios for our configuration. Note that multiple configurations are possible, and it is extremely important to test all scenarios under closely monitored conditions first on the bench but also *in the air* before relying on them for any kind of serious mission.

A. MAVLink BEHAVIOR

Before we begin, it is important to note that MAVLink commands are interpreted as they come. For example, if one control method (e.g. radio control) sends a MAVLink message "change from flight mode A to flight mode B", the craft transitions to mode B. If another control method (e.g. internet control) sends another command such as "change from flight mode B to flight mode C", the craft will change to mode C, even if the switch on the radio still has mode B selected. In particular, as the control methods come in and out of connection, and as the craft goes in and out of failsafe mode, the flight controller always responds to incoming MAVLink mode change messages, regardless of the link status and failsafe status. To be clear, when in failsafe/RTL mode, if either of the control methods (radio or internet) sends a MAVLink mode change message, the craft goes out of failsafe mode and into the new commanded mode.

B. HOW DOES THE PILOT KNOW THE CRAFT HAS ENTERED FAILSAFE?

Failsafe means a lost link. If there is only one link, then the lost link means the pilot does not know the craft has entered failsafe, other than the visual lack of response to commands. In our case, there are three methods for failsafe notification:

1) FAILSAFE NOTIFICATION THROUGH THE RADIO CONTROL

As mentioned above, the TBS crossfire has telemetry built in. Although we do not use it for MAVLink in this paper, it does provide RSSI telemetry to the radio on the ground from the radio receiver in the craft. In the event the link is lost, the radio emits an audible "RSSI critical" verbal warning. Although this is not the same as failsafe event on the flight controller, it does signify to the pilot the RF link is lost or is about to be lost.

2) FAILSAFE NOTIFICATION THROUGH THE GCS

The Mission Planner application on the GCS handles multiple failsafe events, but if internet connectivity is lost, Mission Planner cannot report the craft behavior. On the other hand, if the internet connectivity is still available, it gives an audible "failsafe" warning. Also, the words "FAILSAFE" appear in large bold font over the artificial horizon indicator.

3) FAILSAFE NOTIFICATION THROUGH THE ANALOG VIDEO HUD

The OSD displays "failsafe" in bold at the top of the screen if the flight controller enters failsafe for any reason.

C. RF FAILSAFE (LOST RADIO CONTROL LINK)

In the event that the RF signal is lost, the on-board RF receiver is pre-programmed to send a low-throttle PWM signal (less than 1 ms pulse width, that does not happen during normal flight). (Typically (but not always) this is interpreted by the flight controller as failsafe event; see below). On regain of the

RF signal, the on-board receiver again sends PWM signals to the flight controller as directed by the radio.

In the event that the RF signal is lost, the on-board receiver also sends control surface commands to the flight controller under lost link conditions; these are set by default at neutral.

In the event that the RF signal is lost, note, the radio also has one channel programmed as ''mode select'', which the on-board receiver does send to the flight controller on RF lost link conditions. This can be important; see below.

(It is possible to have a separate ''failsafe'' signal sent over the SBUS from the receiver to the flight controller, but we have found this feature poorly documented and unreliable.)

It is up to the flight controller to interpret these under the scenarios we discuss next:

1) RF FAILSAFE WHEN UNDER RADIO CONTROL

In the case of lost RF link, *when under radio control*, the flight controller interprets the low throttle (pulse width on throttle channel less than 1 ms) as a failsafe, and sends the craft automatically to RTL (return to launch). When terrain following is enabled (see terrain following section above), the craft flies at a fixed altitude above the terrain to the launch/home point, where it circles at a pre-defined altitude, typically 200 ft. If terrain following is not enabled, the craft returns home at a pre-programmed altitude above the home point, which is established at arming.

Once the link is re-established, failsafe is disengaged and the radio transmitter resumes control. Alternatively, the internet control can be used to dis-engage failsafe and continued to control the craft (manually or in autopilot waypoint mission mode).

2) RF FAILSAFE WHEN UNDER INTERNET CONTROL

While under internet control, the craft can be in ''joystick'' control mode or ''non-joystick'' control mode. The user can toggle between ''joystick'' and ''non-joystick'' mode in the Mission Planner application running on the Win10 GCS. The failsafe response to the two cases is different:

a: RF FAILSAFE WHEN UNDER INTERNET CONTROL IN JOYSTICK MODE

In ''joystick mode'', a USB joystick replaces the RF controls. These must be configured in advance, but typically the USB joystick ''sticks'' control the same as the radio, i.e. throttle, pitch, roll, yaw. Additional buttons can be pre-programmed to change modes.

In the case of lost RF link, *when under internet ''joystick'' control*, the low throttle output of the RF receiver is *ignored* by the flight controller, and no failsafe event is registered. *However,* the RF receiver *does* send the ''default no RF signal'' ''mode'' selection, which *does* change the flight mode. The pilot must decide in advance which flight mode this is to be. For example, it could be RTL, so that on lost-link the RTL condition is enabled.

Under RF lost link, the pilot has full internet control of the craft.

When the RF link is restored, the same happens: The radio sends a ''mode select'' message of whatever the radio switch is set to.

Under RF restored link, the pilot still retains full internet control of the craft.

b: RF FAILSAFE WHEN UNDER INTERNET CONTROL IN ''NON-JOYSTICK'' MODE

In the case of lost RF link, *when under internet ''non-joystick'' control*, the low throttle output of the RF receiver is *acknowledged* by the flight controller, and a failesafe event is registered. The craft enters RTL mode.

Again, the pilot can send a mode change command and therefore regain internet control even when the RF link is lost, and continue to pilot the craft as before the RF link lost event.

When the RF link is restored, the ''failsafe'' event is cancelled, and the craft can continue to be piloted under internet control.

D. INTERNET FAILSAFE (LOST 4G CONNECTION)

A chain is only as strong as its weakest link. There are multiple links which can break the connection between the Win10 GCS Mission Planner application and the flight controller in the air (Figure 7). All of these could be defined as ''lost 4G connection failsafe'' events. The method we have chosen to respond to as ''failsafe'' is the lack of MAVLink packets to the flight controller after a pre-programmed time, e.g. 2 seconds. Fortunately, this definition is built into Ardupilot. This has the advantage of allowing the craft to decide how to behave, regardless of which link fails, so the craft does not need detailed knowledge of what is happening on the ground or anywhere else on the internet.

However, the pilot should be aware of the complete chain and monitor it continuously; this added workload in the field may require a dedicated co-pilot for network monitoring, just like in the old days a modern airliner had a third co-pilot as flight control engineer. In the configuration reported here, the network architecture is designed so that it can be managed by a single pilot, but the workload at critical flight points can quite heavy.

We discuss the behavior under each lost 4G connection scenario below:

1) WHEN UNDER RADIO CONTROL: LOST 4G CONNECTION FAILSAFE

When under radio control, loss of internet link is registered as a failsafe event. The craft enters RTL and ''FAILSAFE'' is shown in bold red on the GCS, and an aural warning is given on the GCS. The pilot can continue to fly under radio control by issuing a mode switch command or the pilot may choose to allow the craft to continue to fly in RTL mode.

2) WHEN UNDER INTERNET CONTROL: LOST 4G CONNECTION FAILSAFE

While under internet control, loss of internet control results in a failsafe/RTL condition, regardless of whether in ''joystick'' or "non-joystick" mode. The radio can change the mode to any other mode, which disables the failsafe and re-enables the radio control.

E. INTERNET FAILSAFE CANCEL PROTOCOL: (REGAINED 4G CONNECTION)

For the radio control, reconnection is seamless and automatic. This may not necessarily be the case for the 4G internet connection, due to its complexity, so more discussion and user input may be required here.

The hardware and software internet links have many possible points of failure. An internet connection is a frail thing, with no guarantee of constant on success. It is possible to control all the internet network hardware from craft to ground control station but this is prohibitively expensive for all but the most critical missions such as military, and defeats the point of exploiting the huge investment in wireless infrastructure, easily totaling billions of dollars and covering a large portion of the Earth. The point of this work is to exploit existing investment in 4G cellular infrastructure and internet in general, without requiring a guarantee of connectivity. The following possible "weak links" in the chain exist:

4 computers, each of which may crash:

1. Flight controller
2. Onboard computer (Raspberry Pi Zero)
3. Cloud workstation (AWS instance)
4. Ground-control station (GCS) computer

Two IP connections, each of which may go down:

1. On board computer (Raspberry Pi Zero) to cloud workstation (AWS instance)
2. Cloud workstation (AWS instance) to ground-control station (GCS) computer

Up to four SSH encrypted software tunnels, each of which may go down:

1. Pi Zero to AWS for MAVLink traffic
2. Pi Zero to AWS for video cam traffic
3. AWS to GCS for MAVLink traffic
4. AWS to GCS for video traffic
5. In case a third SSH is enabled to log into the Pi Zero remotely while in flight, that is another software link.

These are all the components in the network configuration figure (Figure 7); one must consider failure analysis of every single one of them. Loss of any of these components puts the craft into lost link, and the failsafe.

In order to enable an elegant reconnection, we must consider and analyze the system response to each particular (adverse) event.

1) SELF-HEALING PROPERTIES

Before we discuss each adverse event, we enumerate the self-healing properties and features of the network. There is only one other such self-healing architecture described in the literature (Zerotier), which is discussed in detail and compared to this work in the "Significance" section below.

a: SELF-HEALING OF THE INTERNET PROTOCOL

Each network device automatically self-connects to the internet and self-heals if a link goes down and comes back up at the hardware level. This is built into the IP protocol.

b: SELF-HEALING OF THE MAVLink PACKET HANDLING CODE

The two Linux PCs (Pi Zero and AWS) run MAVLink Router. This program maintains listen/talk abilities even if the rest of the components fail. It will automatically run as a daemon as long as the Linux OS it is running on is up.

c: SELF-HEALING OF THE PCs

The two Linux computers have scripts to auto-reconnect SSH and to auto-reconnect IP on boot/reboot.

d: SELF-HEALING OF THE SSH TUNNELS

The AWS to FC SSH tunnels run the auto-SSH package which automatically maintains and reestablishes the SSH link if it goes down.

SSH tunnel connect/reconnect: On boot (and reboot) of both Linux computers (the Pi and the AWS server), a script is run which establishes the SSH tunnel. This script uses the "auto SSH" command, which will re-establish the SSH if it is lost, for example if the IP connection goes down and is restored. The other side of the connection (GCS to AWS) must be manually monitored and established, since Windows 10 does not provide the auto-SSH package.

2) COMPUTER REBOOT AFTER COMPUTER CRASH

These are generally rare. In our experience, the only system ever to crash under flight conditions is the Win10 system, which is on the ground.

a: FLIGHT CONTROLLER CRASH AND REBOOT

If this happens, it is catastrophic for the control system. On crash of the flight controller, the throttle will receive no signal and therefore go to idle, and the control surfaces will receive no signal, and the craft will gently glide and land.

b: PI ZERO CRASH AND REBOOT

If it does not reboot, link is lost permanently to the internet and radio control is required for a controlled landing. The craft will enter RTL and return home at which point the pilot can manually land or continue radio control only flight.

It the Pi Zero does reboot, it executes a script to automatically redial the AWS and re-establish the SSH tunnels. These tunnels can be monitored by keeping a terminal to the AWS station open and monitoring all SSH connections.

c: AWS CRASH AND REBOOT

If it crashes, it can be monitored and rebooted from the GCS to re-establish link. The user must manually redial the AWS from the Win10 GCS to re-establish the SSH tunnels.

Unfortunately, in Win10 the ''autoSSH'' package is not available like it is in Linux.

d: GCS CRASH AND REBOOT

If the Win10 PC or Mission Planner application crashes, they can be monitored and rebooted on the ground. SSH tunnels to AWS must be re-established manually (see above) and the Mission Planner must be manually reconnected to the craft.

3) IP RECONNECTION

a: AWS TO GCS

If the IP connection is lost, it will disconnect the terminal log into the AWS, which should be open and monitored continuously during flight. The reconnection must be established manually at the GCS.

b: AWS TO FC

If the IP connection is lost, it will break the SSH tunnel, which can be monitored in a terminal which should be open and monitored continuously during flight. On reconnection, it is self-healed and the SSH tunnel will reappear in the terminal monitored.

At the IP level, the re-connection from the Pi 4G modem to the internet is seamless. The Pi may get a new IP address but the SSH architecture ensures a reconnection.

4) SSH TUNNEL RECONNECTION

Note in both SSH tunnel reconnection cases, the MAVLink Router software package elegantly self-heals and automatically resumes MAVLink package trafficking as before the lost SSH links.

a: AWS TO GCS SSH TUNNEL RECONNECTION

If the SSH connection is lost, it will disconnect the terminal log into the AWS, which should be open and monitored continuously during flight. The reconnection must be established manually at the GCS.

b: AWS TO FC SSH TUNNEL RECONNECTION

The AWS to FC SSH tunnel can be monitored via the terminal to the AWS during flight. The auto-SSH package will self-heal and reconnect the SSH tunnel once lower level hardware and software connections are re-enabled.

F. SUMMARY

The radio control and internet control comprise two redundant control systems. If either one (or both) goes down, the craft enters RTL mode. If the either is still active, it can be used to continue to pilot the craft remotely. If either is re-engaged in flight, it can be used again. Note other failsafe behavior options can be defined in Ardupilot.

This makes sense because the most likely reason for one or the other link to go down is loss of RF signal. If the craft reverses course and flies back into the region with signal, it will reconnect that link. Most of the network is self-healing under such a scenario, but close monitoring and some manual network inputs may be required.

G. TESTING

The pilot can create radio lost link and test the failsafe behavior simply by powering down the radio control. The pilot can create an internet lost link by either 1) rebooting the AWS server remotely, or 2) logging into the Pi and rebooting it (even while in the air). All these have been tested (first on the bench, the in flight) to confirm the expected behavior.

IX. USE CASE: CAN YOU HEAR ME NOW?

This class of UAVs can be used to efficiently map wireless network coverages over large areas at various altitudes, something which is missing from existing coverage mapping technologies. Most mapping technologies use ground based vehicles, which cannot access remote locations, where coverage maps are most likely to be at the border of cell tower reach. Computer modeling is used primarily, but there is no substitute for physical verification.

A. PROOF OF CONCEPT

In order to demonstrate proof of concept of this idea, we flew the craft over a remote, unpopulated area until it reached the edge of the cell tower coverage. In this simple proof of concept case, an internet disconnect failsafe (see above) was used as a binary ''go/no go'' to map coverage. In this test flight, for safety, radio control link was maintained when the internet connectivity was lost (see Figure 8).

At the edge of the tower coverage, the craft hit return to home failsafe due to lost internet connectivity, reversed course, and headed home. Using the radio control we purposefully flew the craft to the edge of coverage three times, and each time it reach a specific geographic point, it lost internet connectivity and failsafe/RTL was automatically engaged. This was reproducible and occurred at the same physical point in space within less than 200'. This clearly demonstrates proof of concept that this class of UAVs can be used to map cellular coverage in areas inaccessible to vehicles or pedestrians.

B. POSSIBLE IMPROVEMENTS

In principle the RSSI and quantitative signal strength can be mapped to for three dimensional coverage databases. These can be uploaded to the cloud in real time, where big data processing can be used to incorporate computer models of coverage into actual measured data. This would require access to the modem raw data, which Verizon does not publicly provide, but custom measurement equipment can be built for this purpose. This could result in significant savings for wireless communications companies and provide enhanced accuracy of wireless coverage maps, as well as enhance models of wireless propagation in ways previously not possible with only ground based measurement data, including models of UAV network connectivity.

C. PRIOR ART

Prior literature on this concept exists, but is mostly based on simulations, numerical modeling, and proposed architectures (such as drone swarm architectures and topologies for internet distribution) that have not yet been realized [38]–[66]. At this point we estimate that there are more academic papers on communications and ‘‘drone swarms’’ than the total number of drones that have actually ever flown in swarms! The reason is that the drones/UAVs (until now) had limited flight time and significant safety and regulatory hurdles. It is our hope that through this paper and this new class of UAVs that we can move the needle towards demonstration and deployment, so that the field can move forward from pure academic, almost speculative literature to actual use cases.

The first cellular (GSM) network connected UAV was demonstrated by Sweden in 2006 [67]; the UAV weighed over 95 kg and the bitrate was around 10 kb/s. A prototype multi-rotor version of antenna mapping was demonstrated in 2016 the authors' lab [68], measuring for the first time with a drone the antenna radiation pattern of a cloverleaf, dipole, and patch antenna. Subsequently, Qualcomm published a much more extensive trial in 2017 using a drone to study LTE link quality [69]. Both our study and the Qualcomm study used a quadcopter with $\sim$ 1 kg weight and 15 minute flight times, almost 4x more weight and 4x less flight time compared to the work presented in this paper. In 2016 KU Leuven studied the effects of LTE signal propagation and interference at different altitudes using an LTE receiver, using a manned aircraft in lieu of a UAV to measure signal behavior [70]. In 2016 [71] a hobbyist provided a bare bones description of how to integrate a 4G modem with a DJIS900 octocopter airframe (system weight 4.7-8.2 kg (ref dji spec) for telemetry (not video), although no flight demonstrations were reported. In 2018, Ericson presented models and data supported by measurements of LTE signal propagation using a manned helicopter [72]. LTE signal loss models were measured by a Danish group in 2017 [73] using a DJI Matric 600 (weight 10 kg). In 2018 a group in Australia [74] presented LTE signal loss data as a function of depression angle using a 3DR Solo quadcopter (weight 2 kg; flight time 15 mins.) A large $\sim$ 2kg custom quadcopter was used to measure 900 MHz signal propagation by NASA in 2017 [75]. In 2017, the Universitat der Bundeswehr Munchen demonstrated public LTE control and video using a VPN of a 6 kg octocopter with 30 min flight time [76]. In 2018, RobSense technology demonstrated a proprietary Lora based hardware/software UAV control platform, but as a side note in the work (with no details on the actual UAVs used), showed that 4G connections were better suited for a UAV swarm of 10 UAVs [77], with another abstract by the same group in [78].

The Third Generation Partnership Project (3GPP), an industry consortium on cellular networks including 5G and beyond, initiated a study on enhanced LTE support for UAVs in March 2017 [72] and refs. therein. Ericson, AT&T, China and others have recently gotten into the game. Many times a press release is given whose only purpose is click bait or to glorify a particular brand, company, or organization (a common practice in the world of public relations) to get news views and no technical details are divulged ([79]–[81]).

In sum, there is huge interest from industry in the use of UAVs to enhance existing coverage models, to provide additional coverage through mobile aerial base stations, and to enhance the network for the purpose of improving UAV networking. However, to date in spite of enormous recent flurry of academic publications and significant industry investment in measurements using manned vehicles, much remains to be done. This new class of micro-UAVs is poised to dramatically improve these spotty initial measurements, verify sophisticated models, and to do so at a much lower cost and safer environment than any other class of UAV.

X. SAFETY AND SECURITY

Safety of UAVs is a critical concern given that there are more registered commercial UAV pilots (both commercial and hobby) in the US alone as compared to manned aircraft pilots (Table 5), and probably ten to a hundred times more unregistered UAV pilots. We discuss some safety advantages of this class of UAVs.

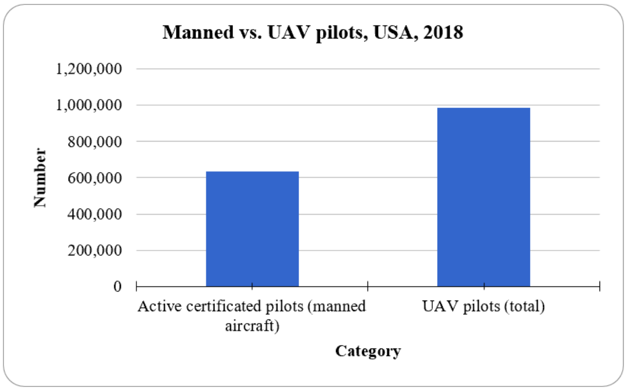
Because most of these one million new pilots do not have a prior background in aviation safety, we discuss later in this article the importance of pilot attitudes and pre-requisite knowledge and skills for safe and effective deployment of the technology described in this article.

At the moment, there are three main ‘‘definitions’’ of safety: 1) Scholarly, peer-reviewed, detailed studies of collision damage between UAV and manned aircraft/ people [84]–[86], 2) Non-peer reviewed blog/youtube/press release/social media discussions [87], [88] (which for better or worse play out in the court of public opinion, and may be completely correct in many cases [89], [90]) and 3) Governmental regulations. From a scientific point of view, #1 is the most important. In modern democratic societies, #2 forms the basis for policy since voters ultimately decide the law. #3 in principle should reflect #1 and not #2. Research universities, as the trusted public institutions of knowledge and education, have the moral duty to reconcile #1 and #2, which they do no always succeed at.

For 1), there are very few studies regarding UAS safety. The ASSURE group (one of the FAA's Centers of Excellence for UAVs), funded with $\sim$ \$3M from the US federal government, and $\sim$\$18M in total funding including matching funds, has embarked on a large scale study [91], reviewing the safety literature on collision damage models (over 300 articles, many pre-dating the UAV era), and performing its own quantitative tests and analysis [92]. Phase 2 of the final report is not available as of this writing [93], but 3) this has not prevented the FAA from issuing a notice of proposed rulemaking (NPRM) to define ‘‘safe’’ as far as UAV flights over people are concerned [94]. Since this final study is not yet available, we will refer this work to the FAA proposed

TABLE 5. Manned vs. UAV pilots in the USA, 2018 [82], [83].

Category	Number
Active certificated pilots (manned aircraft)	633,317
UAV pilots (total)	984,321
FAA Part 107 (commerical)	106,321
Hobbyists	878,000



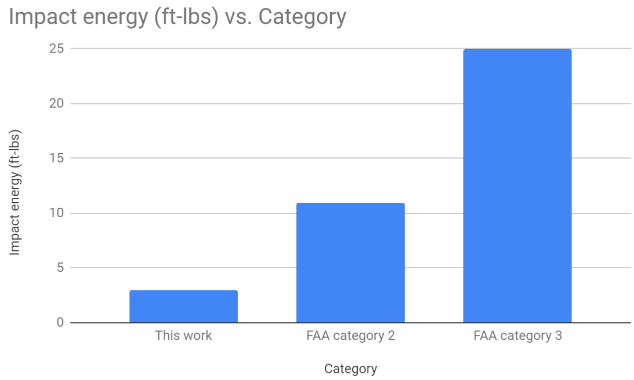


FIGURE 16. Impact energy vs. FAA defined safety category. FAA safety category 1 only defines UAV weight of 250 g. In this work, we have a weight of 300 g, and assume a cruise speed of 30 mph for impact, giving impact energy of 3 lb.-ft. For this particular airframe, a throttle off stall speed is 20 mph, which would give an even lower impact energy of 1.3 lb-ft., over 10x lower than the FAA defined safety threshold.

definition of safety, as well as provide simple physics based analysis, for the airframe studied in this work.

A. FAA DEFINITION OF SAFETY

While the first version of this article was under review, the FAA announced that it has defined 250 g as a "class one" safety level, which can be safely flown over people. It defined additional classes at higher weights which must be certified, and categorize the UAV in terms of impact energy on collision. Analysis shows that this paper's airframe falls into the safe category as defined by FAA (Figure 16).

For category 2, FAA requires the propeller not cause laceration. In this airframe, the propeller is in the back. Unless the plane somehow ends up flying backwards prior to impact (an extremely unlikely scenario given the aerodynamic glide and self-leveling tendencies of this airframe), the propeller would not cause laceration.

Although not entirely scientific, a recent youtuber (Bruce Simpson [90]), confident of the safety of 250 g foam UAVs on impact with humans, did what Bayer chemist Eichengrum did when he invented aspirin [95]: He tested it on himself. Although we highly advise against this general method for both Bayer chemists and UAV pilots, the result in both cases was a success. For Bayer, we have aspirin. For UAVs, a 250 g foam wing with hidden propeller caused no damage to human or craft after multiple collisions. Note that for quadrotors, a different class of UAVs, even a 250 g craft could cause severe laceration if the propellers were not shielded. This justifies our claim that this is a new class of UAVs, safer and distinct from quadcopters of the same weight and size.

B. SECURITY

Since the webcam, telemetry, and command and control of the UAV are transported over the open internet, security of the data is critical. Because all of the internet traffic is encrypted with the SSH protocol, which is generally trusted to be secure [31], the overall system is secure. It is even more secure than traditional UAV control methods, which can be spoofed [96], [97] allowing a remote operator to take control of the bird. No such spoofing is believed to be possible with SSH. Further discussion of the security of the SSH protocol is outside the scope of this article, and the reader is referred to the security literature for additional discussions on that topic [31].

C. PRIVACY, INVASIVENESS, NUISANCE

At over 200' this class of UAV is virtually impossible to hear. At lower altitudes, it is very quiet, and it does not have the capability to loiter near people or property. These features inherently enable this class of UAVs to be free from public fear and concern regarding nuisance, invasiveness, and privacy.

XI. SIGNIFICANCE

What is new about this work (the original research portion) and this review of prior classes of UAVs? There are many new things presented in this paper, not just one.

The primary new technical accomplishment in this paper is the system assembled from open source hardware and software components that is lighter (and therefore safer than) any prior 4G UAV, as well as having the longest demonstrated flight time, over an hour. The components and software are mostly COTS, so the technical advance is the integration into a light UAV. Heaviver UAVs have already been demonstrated with this same functionality (Figure 17). Because most solutions are proprietary or poorly documented (if at all), it is hard to compare line by line this UAV to other commercial UAVs in terms of components, but the paper clearly spells out the entire build and programming process and provides an online manual (wiki) for flight operation as part of the supplemental information (www.gitlab.com/pjbca/4guav/wiki),

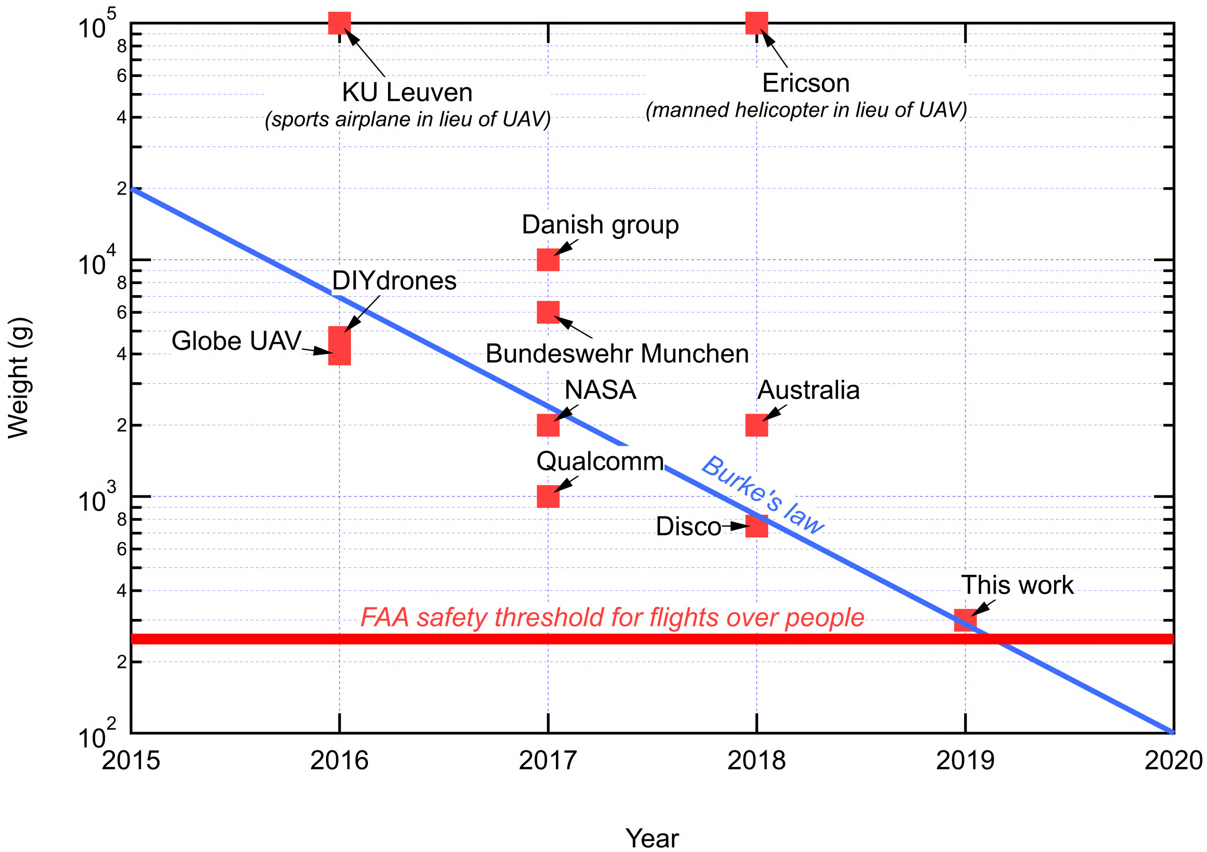


FIGURE 17. **4G connected UAV weight vs. year. References in text. The proposed "Burke's Law" would predict a 100 g 4G connected UAV in the year 2020 or 2021. Note the FAA defined safety threshold is "fuzzy", in that UAVs heavier than 250 g can be considered safe for flights over people if the impact energy is below a certain threshold and the propellers are protected, both of which are satisfied by the class of UAVs presented in this work.**

which no other paper in the academic literature on UAVs does (to our knowledge) for *any* class of UAVs. This simultaneously removes two critical pain points for the massively parallel scaled deployment of these UAVs: First, and most importantly, operator awareness and training has been lacking in drones and caused a global governmental overresponse in regulations. Even the National Academy of Engineering has taken this position in its recent report, which states the governmental regulations of drones are too strict and do not properly [98], [99] weigh the cost/benefit analysis of risk/benefit. Second, almost all UAV system designs are proprietary or passed on verbally from faculty to students in academic university settings or mentor to mentee in non-academic settings. This prevents newcomers to the field from contributing, tinkering, and researching, and in particular has been a HUGE barrier to entry for 4G modem connections. Thus, this paper kills two proverbial birds with one stone and takes a huge step in removing these barriers to entry for current and future UAV engineers, pilots, governmental regulators, and researchers.

A second technical accomplishment is the develop of an open source, self-healing internet architecture and the associated failsafe protocols in case of lost link. The technical accomplishment from a software engineering perspective is to allow a centralized cloud server to coordinate the airborne and ground computers and software which may sit behind firewalls, and to do so in a self-healing way. The architecture was proposed in DIY blog [100] but its implementation was not described in detail, and the self-healing aspect was not proposed there either. This architecture is not in any academic literature we are aware of, and never in any detail on any diy blog or forum, and the source code to set it up has been developed based on the Linux platform and posted in an online repository (gitlab.com/pjbca/4guav). However, as with most Linux projects, it stands on the shoulders of giants such as Linus Torvalds [101] and countless other Linux developers who have already established most of the code needed to implement the technical solution described herein.

So it is both a new architecture and its actual implementation for UAV cellular to ground control that has the following qualities:

- Self-healing (in case of lost link)
- Encrypted (hence secure)
- Able to tunnel through multiple firewalls
- Open source

In addition, it is actually demonstrated in flight hardware in the field rather than proposed. There is only one other published architecture which has these properties (Zerotier, an Irvine, California based startup, described in more detail below). The other option is a VPN, which is much more complex than our solution, but possible [76].

The open-source nature is a very big advantage of this approach, not only for applications of single UAVs, but for more sophisticated applications involving entire fleets. Since eventually UAVs will need to have intelligent responses to integrated traffic management systems, the use of a fractured and proprietary set of UAV avionics software systems will be a disadvantage. It would be as if the internet had multiple communications standards and only subsets of computers could communicate with each other from a specific company. In this paper, due to the open source nature, based on the IP protocol, well established Linux OS, and open source security protocols such as SSH, the path towards an integrated air traffic management system with intelligent drones is smoother. It is similar to the vast majority of servers running Linux that powers the internet today. Ultimately, for fleets of drones to share the airspace, some sort of open-source communications will be required.

Finally, this paper shows how this is a new class of UAV, distinct from other classes, and lists its advantages in comparison to the rest of the UAV industry, namely:

- Light weight (300 g vs. many kg) hence safety (Figure 17)
- Long flight time (hour) (Figure 18)
- Hidden propeller (as compared to quadcopters)
- Cost

This is perhaps the most important point of this article, which is that, after the technical accomplishments #1,2 which are new and novel, an entire new class of UAVs is now available to the research community and the industry. The paper describes the industry overview and clearly puts this class in perspective.

A. PRIOR ART ON 4G CONNECTED UAVs

It is well-known that there have been several attempts to add 3G/4G to UAVs, and in fact, many solutions are already available and on the market. Recent papers have also pointed out this need [46]–[48], [78], [102], and the prior art section above summarized the research papers that have shown this over the last 10 years. In Figure 17, Figure 18 we plot weight vs year and flight time vs year showing our result in perspective and the FAA "safe" limit of weight. Just like Moore's law for integrated circuits [103], we propose a "Burke's Law" for drones: Size of 4G connected drones shrinks by 10x per 2 years. This would predict a 100 g 4G connected UAV in the year 2020 or 2021.

1) COMMERCIAL HARDWARE/SOFTWARE SYSTEMS

In addition to publications, there are some commercial solutions on the market. This includes Skydrone [104],

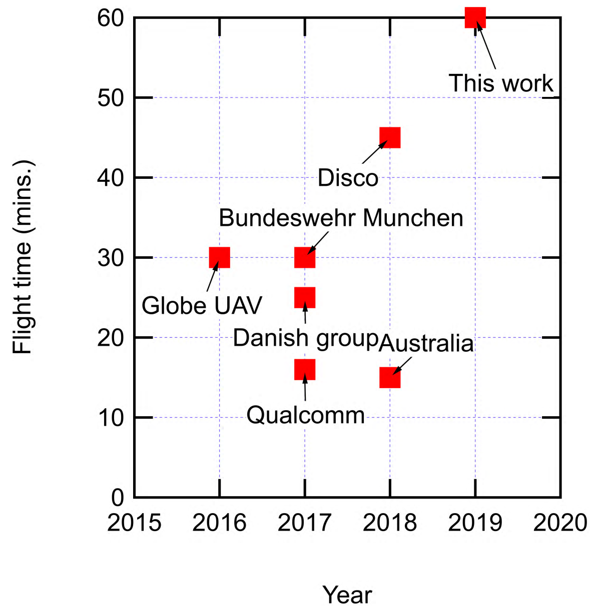


FIGURE 18. Flight time vs. year for 4G connected UAVs. References in text.

which offers a proprietary ~$500 camera/computer hardware combo weighing ~250 g (90 g without case), and proprietary software suite on both ends (ground, air). This compares to the open source hardware/software in this paper which weighs in at 10 g costing $25 for camera/computer. Both solutions require a 4G modem which in our case is a 15g 4G Verizon modem. Obviously, our work proves that miniaturization to a "safe" weight is more than possible, proving this class of UAVs as distinct. However, clearly additional miniaturization is possible by integration of the components (modem, on board Linux computer, and camera). This would make this class of UAVs as a platform even longer lasting, lighter, and safer.

GlobalUAV [105] offers an integrated UAV octocopter system with weight of 4 kg or larger (depending on payload), flight time of 30 minutes, and 4G connectivity for video and telemetry.

2) COMMERCIAL SOFTWARE SYSTEMS

In our opinion, the most challenging aspect of UAV use is the safety issue for use cases involving long-distance, beyond line of sight, and long flight time platforms. We believe we have demonstrated the most successful platform technically on all of these metrics in this paper. It is only when these challenges are really met in the eyes of regulators, scientists/engineers, and the general public that UAVs can really be deployed at scale and evolve to their true potential. Once that happens, there will be a huge demand for back-end software management of the fleet and all of the data (big data) that it presents. Several companies have already taken this back-end step, but these are of only limited utility at the moment as the number of UAVs controlled is small compared to the ultimately envisioned scale. We now discuss these back-end software companies:

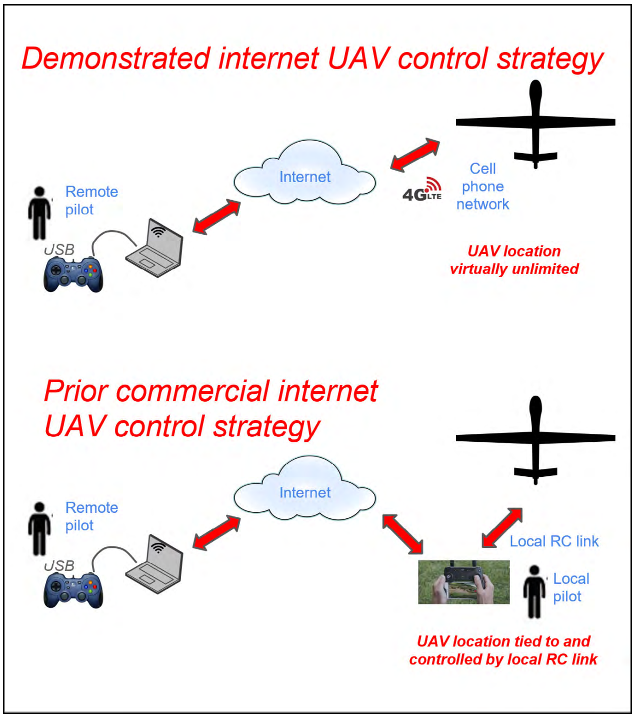


FIGURE 19. A) In this and related 4G UAV works, the location of the UAV is technologically virtually unlimited, even though existing regulations do not yet allow this mode of operation in many cases, and require a local safety pilot as backup (not shown). B) Commercial approaches such as Cape [107] and Flythere [108] rely on local RC control for UAV command and control, restricting the UAV to a local pilot or ground station.

Flytbase [106] offers a proprietary, subscription based, high level cloud based drone management software suite, including a web based SDK and APIs for interfacing to various hardware and firmware. However the hardware and firmware interface seems to be left to the user to develop, or the user must contact the company to contract for a custom solution. They do offer one hardware option, which is a Rasberry Pi 3 based starter kit without 4G modem or video for $399.

Cape [107] and Flythere [108] provide software services using DJI drones which connect the drones local remote controller to the internet through the attached tablet PC. This allows remote control of the drone over the internet, but the drone is restricted to be within radio range of the local remote control. Therefore, while it enables an internet interface, in some sense it does not "count" as a truly internet connected drone, because it is the local remote control that is connected to the internet, not the drone itself. Figure 19 presents a visual comparison of the architecture for the 4G UAVs discussed in this paper, vs. the architecture offered by commercial service companies such as Cape and Flythere.

3) ZEROTIER

The problem of moving hardware behind firewalls and NAT servers is not unique to UAVs. In the era of the internet of things, the problem is more ubiquitious. Although our proposed solution works for UAVs and the user has 100% control over his/her data without the ability of any 3rd party to monitor it, there are alternatives. One such innovate alternative is being offered by an Irvine, California based startup company called "Zerotier". Zerotier [109] is a suite of open source software based on a protocol which gives each devices its own "Zerotier" address, similar to how the IP protocol gives each computer its own IP address. However, Zerotier builds on that by allowing all devices to communicate with each other through NAT, firewalls, etc. using encrypted traffic. The user must install the Zerotier client and server software on their machines/devices to enable this functionality, and Zerotier does require a 3rd party server to manage all the addresses, which is controlled by Zerotier or users can set up private Zerotier networks. Developing an IOT protocol which replaces the IP protocol is clearly beyond the scope of this article. However, one software solution built on the Zerotier platform is available for UAVs, and is a proprietary for profit software solution compatible with Raspberry Pi platforms and 4G modems [110], and also proprietary ground control software. The software, according to the docs, is very CPU intensive and uses 50-100% CPU time on the light RP0W used in this paper, and works better on the heavier Rpi3. In contrast, our networking solution PLUS video service uses only 5-10% of the CPU time on the Raspberry Pi Zero W. Because the uavmatrix software is proprietary, we cannot assess the reason for the difference, and whether it is related to the extra overhead of using Zerotier or some other reason. No record of uavmatrix with a lightweight UAV like this paper (300 g) exists, but it also should be possible as an alternative.

A second software modification [111] based on Zerotier has been provided for the now discontinued Parrot Disco [112]. The Parrot Disco is a wing design (weight 750 g, flight time 45 minutes) with proprietary Linux based hardware/software combination ("the puck") based on Ardupilot codebase.

In sum, there is a variety of 4G UAV solutions available in the research literature, as commercial software packages, and even as entire UAV systems. In general, they increase in cost and size (UAV weight) as the system becomes more and more user friendly as a "turnkey" solution. However, none of them has the light weight or long flight time described in this paper.

B. THIS WORK VIS A VIS ARDUPILOT

The Ardupilot code base is mature and has been fielded on over one million vehicles [5]. The use of the Ardupilot code on a low cost, lightweight, integrated flight controller with on board OSD and current sensor is new as of late 2018 and codified in this paper, although the developers have also flight tested it on other airframes.

The companion computer to have on board computing is also integrated into Arudpilot, but has never been demonstrated on such lightweight hardware before.

a: ARDUPILOT FEATURES

The Ardupilot software features demonstrated in the "performance" section, namely terrain following, autonomous waypoint missions, auto-takeoff, and auto-landing, have all existed in the Ardupilot codebase for quite some time. They have been demonstrated on much heavier airframes. However, they have never been demonstrated on such a light airframe as this. The developers of Ardupilot have done a wonderful service to the open source community by porting the entire codebase to Chibios. This allowed the Ardupilot software to be abstracted from the hardware, and enabled much lighter flight controllers (with integrated OSD and current sensing hardware) to be used, and enabled this low weight demonstration. The developers have really opened the Ardupilot codebase to a whole new world of hardware, and the future for this is bright, with new upcoming boards based on F7 and H7 microcontrollers, Linux, and virtually any other board. Therefore, the new thing about the flight performance in this paper is not the software, or those particular features (which have been demonstrated before), but their demonstration for this first time on such a small platform, enabled by the developers of Ardupilot taking the wise but difficult step to make the software "portable" to a larger class of lighter, more integrated flight control hardware, including the ultra-lightweight UAV demonstrated in this paper. This was not possible even one year ago.

C. THIS REVIEW

In [116], an outstanding and unique review on UAVs, the various classes of UAVs are summarized in comprehensive and fine detail, but internet command and control for long range, global reach is not discussed. Reference [117] provides a summary of UAV flight simulators. This paper represents the first review of internet connected UAVs.

XII. PRE-REQUISITE SKILLS FOR REPRODUCTION

We outline skills and experience necessary to reproduce this work in the area piloting, physical construction, and software use.

A. PILOTING

Users should be able to pilot the plane under manual control. The craft should be trimmed for straight and level flight with neutral control inputs from the pilot. This requires a significant amount of practice (stick time) and for a novice is likely best done with the assistance of an experienced remote pilot. In the US, the Academy of Model Aeronautics [118] provides a network of clubs with flight instructors available at over 2,000 such flying sites around the country.

B. CONSTRUCTION

Users should be able to solder wires, and create wiring looms with various connectors. Crimping is not required but a useful skill set.

Users should be familiar with and able to apply the safety issues of Li Ion batteries, which can be a fire hazard if handled, stored, charged, or discharged improperly.

C. SOFTWARE

Users should be familiar with Linux, able to log into a terminal and use the command line, copy and move files, edit simple text files, and be familiar with the SSH protocol. Users should be able to read the Ardupilot documentation completely and understand and digest the flight modes, settings, and configurations, in full detail.

D. PERSONALITY

Users should be prepared for setbacks and demonstrate patience and perseverance. The best approach is to add each feature step by step one at a time and test it (on the bench first then in flight) each time under safe conditions away from people and property.

XIII. DISCUSSION: USE CASES

The use cases and optimum sweet spot of most UAVs are still being defined, with aerial photography the number one application as of this writing. In contrast, for the autonomous and lighter, safer nano UAVs, the use cases of this new micro-class of internet enabled UAVs are not yet apparent. However, the use cases of the airplane were probably not entirely predicted by the Wright Brothers when they invented the airplane in Kittyhawk, North Carolina [119].

One possible example is to provide Wi-Fi access to first responders or victims in cases of natural disasters such as earthquakes and forest fires which wipe out local infrastructure. (A similar proposal exists to use UAVs for enhancing cellular coverage [120].) A fleet of disposable micro-UAVs with 50 mile range and one hour flight time could be strategically pre-deployed and fly to loiter in affected areas providing life-saving information and two-way communications between victims, authorities, and even family members, with very low risk to those on the ground or in the air. The entire continental US could be "insured" with a fleet of roughly 10,000 pre-deployed nano-UAVs. At a cost of $100/UAV (projected at scale), this would only be $1M to cover the entire US. This is not presently possible with quadcopters, which pose greater risk, higher cost, and lower range.

Additional use cases may involve counter-terrorism and threat detection of chemical and biological weapons, since long loiter time, long distance standoff monitoring is enabled. This can be for protection of both civilian and military populations. The use of nanowire sensors on board UAVs is already envisioned by the US Air Force [121].

Environmental monitoring of pollution and air quality is another possible use case [122]. This would enable monitoring of air quality over spatial areas not possible with ground based sensors e.g. wilderness areas with no roads.

Unfortunately, in this case, government agencies are not even considering this class of vehicles. (Note: After this manuscript was submitted, the FAA put out a draft rule to

TABLE 6. Technology classes and performance of various UAVs [2], [13]–[15], [113]–[115].

Technology	Model/Class		Weight (g)	Ctl range (miles)	Flight time	Cost	Flt range (miles)
Heli	Black Hornet		16	1	25 mins.	$43,000	1
Quad	Nano quad class		25	0.1	3 mins.	$100	0.1
Wing	This work		300	infinite	1 hour	$400	50
Quad	DJI class		2000	5	30 mins.	$1,000	15
Airplane	Mini Talon class		2000	50	1 hour	$500	50
Airplane	Predator class		10000	infinite	24 hours	$64,000,000	675

allow flights over people for small UAVs under a certain weight and impact force limit [94].) Partly this is because the general public is not educated about this new class of UAVs. However, this is not excusable: Regulatory agencies such as the FAA should be held to a higher standard. In particular, they should not take a knee-jerk, uneducated approach to this new class, and treat them the same as 2 kg quadrotors. To do so could permanently halt progress in this nascent field, at least in the US. Most importantly, US universities should encourage this development. Academic administrators should not pile on additional regulations above already onerous federal regulations. Such additional restrictions, primarily out of ignorance of the administrators, although well meaning, completely contradict the mission of the American modern research university, one of the best features of American society recognized and envied around the world. And faculty and students in the field should do their part to educate the powers that be. With the breakneck pace of technical innovation, emerging public impact and opinion, and governmental and industry growing involvement, students and researchers in this field will be very busy for some time to come!

XIV. SUMMARY

This class of micro-UAVs which are cheap, safe, and secure provides yet another paradigm for unmanned aviation, in an already fast moving technological field (XI-C). The first revolution was the 2 kg quadcopters which, although functional and easy to use, are limited in flight time to 30 minutes and provide significant risk of damage or injury in case of malfunction. A second class are nano-UAVs with very limited flight time (few minutes) and short range. This intermediate provides long range, safe glide (does not fall from sky) in case of hardware failure, long flight time, and low cost, and provides internet control, telemetry, and HD video over an encrypted secure channel. This in essence defines a completely new class of UAVs.

APPENDIX

Multiple additional material is available. The source code for the Linux networking configuration developed and described in this work is available in an open-source online repository, together with a wiki (including detailed install instructions, as well as an operations field manual) on its configuration, at: www.gitlab.com/pjbca/4Guav.

Flight footage from the multiple cameras (analog ground station with HUD, analog HD footage from onboard SD recorder, digital recorded video) is available at the author's Youtube channel https://www.youtube.com/channel/UCS6W3xsDm8ie6Y1YjUxrQ4A; Including specifically: internet control and terrain following: https://www.youtube.com/watch?v=mSEyjjQ7Iks https://www.youtube.com/watch?v=gU2C5MIcSfU; self-flying-airplane missions (auto-takeoff, waypoint missions, auto-land): https://www.youtube.com/watch?v=w20KZrZFU14 and https://www.youtube.com/watch?v=8wlufbZMPig

Failsafe and self-healing internet connection demonstration in flight: https://www.youtube.com/watch?v=82nAR358jMs

The configuration file for the Ardupilot Flight Controller is available upon request.

Open-course code that was developed prior to this work but used in this work is also available at various open-source repositories: All of the Ardupilot source code, Mission Planner, as well as detailed documentation is available at https://github.com/ArduPilot/ardupilot. The MAVLink Router source code for the Linux controlled routing of MAVLink packets is available at github at: https://github.com/ArduPilot/MAVLink Router. The webcam setup for the Raspberry Pi is available at:

https://github.com/silvanmelchior/RPi_Cam_Web_Interface. The various Linux encryption packages (such as auto-SSH) are available from the standard Linux sources.

ACKNOWLEDGMENTS

The opinions expressed herein are those of the author and not necessarily of the University of California system.

REFERENCES

[1] G. M. Simons, *Howard Hughes and the Spruce Goose: The Story of the H-K1 Hercules*. Barnsley, U.K.: Pen and Sword, 2014.
[2] M. Streetly, *Jane's All the World's: Unmanned 2018-2019*. Coulsdon, U.K.: Jane's Information Group, 2018.
[3] *Ardupilot Home Page*. Accessed: Apr. 6, 2019. [Online]. Available: www.ardupilot.org
[4] *Ardupilot Code Base*. Accessed: Apr. 6, 2019. [Online]. Available: www.github.com/ardupilot
[5] *About Ardupilot*. Accessed: Apr. 6, 2019. [Online]. Available: http://ardupilot.org/about
[6] Tridge. *ArduPilot Port to ChibiOS*. Accessed: Apr. 9, 2019. [Online]. Available: http://www.chibios.com/forum/viewtopic.php?t=4463
[7] *Chibios Homepage*. Accessed: Apr. 9, 2019. [Online]. Available: http://www.chibios.org/dokuwiki/doku.php
[8] *Chibios Code Base*. Accessed: Apr. 9, 2019. [Online]. Available: https://github.com/ChibiOS
[9] *Betaflight List of Compatible Boards*. Accessed: Apr. 9, 2019. [Online]. Available: https://github.com/betaflight/betaflight/wiki/Hardware-Reference
[10] *Raspberry Pi Zero W*. Accessed: Apr. 9, 2019. [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-zero-w/
[11] *Raspberry Pi Size and Weight Benchmark*. Accessed: Apr. 9, 2019. [Online]. Available: https://www.raspberrypi.org/magpi/raspberry-pi-specs-benchmarks/
[12] *Amazon Web Services*. Accessed: Apr. 9, 2019. [Online]. Available: https://aws.amazon.com/
[13] Strix. *Nano Goblin Specifications*. Accessed: Apr. 10, 2019. [Online]. Available: https://www.readymaderc.com/products/details/strix-nano-goblin-high-performance-fpv-plank-pnp#features-tab
[14] *Phantom 4 Specifications*. Accessed: Apr. 10, 2019. [Online]. Available: https://www.dji.com/phantom-4/info#specs
[15] *Mini Talon Specifications*. Accessed: Apr. 10, 2019. [Online]. Available: https://www.readymaderc.com/products/details/xuav-mini-talon-pnp#features-tab
[16] *Omnibus F3 Nano Specification*. Accessed: Apr. 9, 2019. [Online]. Available: https://www.readytoflyquads.com/flip-32-f3-omnibus-nano
[17] *Omnibus F7 Specifications*. Accessed: Apr. 9, 2019. [Online]. Available: https://store.myairbot.com/flight-controller/omnibus-f7/omninxtf7.html
[18] *Hollybro PM02 Datasheet*. Accessed: Apr. 9, 2019. [Online]. Available: http://www.holybro.com/manual/PM02.pdf
[19] *Pixhawk 4 Technical Data Sheet*. Accessed: Apr. 6, 2019. [Online]. Available: https://github.com/PX4/px4_user_guide/raw/master/assets/light_controller/pixhawk4/pixhawk4_technical_data_sheet.pdf
[20] *Omnibus F3 Femto Specification*. Accessed: Apr. 9, 2019. [Online]. Available: https://www.readytoflyquads.com/f3-femto-flight-controller
[21] *Omnibus F4 Nano Specifications*. Accessed: Apr. 9, 2019. [Online]. Available: https://store.myairbot.com/flight-controller/omnibus-f3-f4/omnibusf4nanov6.html
[22] *Omnibus F4 Pro Specifications*. Accessed: Apr. 9, 2019. [Online]. Available: https://store.myairbot.com/flight-controller/omnibus-f3-f4/omnibusf4prov3.html
[23] *Arkbird Specifications*. Accessed: Apr. 9, 2019. [Online]. Available: http://www.arkbirdfpv.com/html_products/Arkbird-OSD-37.html#.XK0WZKy6OUl
[24] *Eagletree Microvector Specifications*. Accessed: Apr. 9, 2019. [Online]. Available: http://www.eagletreesystems.com/index.php?route=product/product&product_id=153
[25] *Eagletree Vector Specifications*. Accessed: Apr. 9, 2019. [Online]. Available: http://www.eagletreesystems.com/index.php?route=product/product&product_id=136
[26] *Pixhawk 2 Datasheet*. Accessed: Apr. 9, 2019. [Online]. Available: https://docs.px4.io/en/flight_controller/pixhawk-2.html
[27] *MinimOSD Datasheet*. Accessed: Apr. 9, 2019. [Online]. Available: http://www.holybro.com/manual/MicrominimOSDManualv1.0.pdf
[28] *History of Ardupilot*. Accessed: Apr. 9, 2019. [Online]. Available: http://ardupilot.org/copter/docs/common-history-of-ardupilot.html
[29] T. Blacksheep. *TBS Crossfire Nano Specifications*. Accessed: Apr. 9, 2019. [Online]. Available: https://www.team-blacksheep.com/products/prod:crossfire_nano_rx
[30] *Verizon USB730L 4G Modem Specifications and Data Sheet*. Accessed: Apr. 9, 2019. [Online]. Available: https://ss7.vzw.com/is/content/VerizonWireless/Devices/Verizon/Userguides/vzw-global-modem-usb730l-ug.pdf
[31] D. J. Barrett, D. J. Barrett, R. E. Silverman, and R. Silverman, *SSH, The Secure Shell: The Definitive Guide*. Newton, MA, USA: O'Reilly Media, 2001.
[32] *Mavlink Developer Guide*. Accessed: Apr. 11, 2019. [Online]. Available: https://mavlink.io/en/
[33] Intel. *Mavlink Router*. Accessed: Apr. 11, 2019. [Online]. Available: https://github.com/intel/mavlink-router
[34] OpenTX. *Welcome to OpenTX*. Accessed: Apr. 11, 2019. [Online]. Available: https://www.open-tx.org/
[35] M. Osborne. *Mission Planner*. Accessed: Apr. 11, 2019. [Online]. Available: http://ardupilot.org/planner/
[36] *Terrain Following*. Accessed: Apr. 19, 2019. [Online]. Available: http://ardupilot.org/plane/docs/common-terrain-following.html
[37] T. BlackSheep. *TBS Crossfire—100 km Range Test -UHF Control Link*. Accessed: Apr. 11, 2019. [Online]. Available: https://www.youtube.com/watch?v=ULVwMSL5xac
[38] L. Gupta, R. Jain, and G. Vaszkun, "Survey of important issues in UAV communication networks," *IEEE Commun. Surv. Tuts.*, vol. 18, no. 2, pp. 1123–1152, 2nd Quart., 2016.
[39] X. Zhou, S. Durrani, J. Guo, and H. Yanikomeroglu, "Underlay drone cell for temporary events: Impact of drone height and aerial channel environments," *IEEE Internet Things J.*, vol. 8, no. 2, pp. 1704–1718, Apr. 2018.
[40] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal LAP altitude for maximum coverage," *IEEE Wireless Commun. Lett.*, vol. 3, no. 6, pp. 569–572, Dec. 2014.
[41] A. Al-Hourani, S. Kandeepan, and A. Jamalipour, "Modeling air-to-ground path loss for low altitude platforms in urban environments," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2014, pp. 2898–2904.
[42] S. Rohde and C. Wietfeld, "Interference aware positioning of aerial relays for cell overload and outage compensation," in *Proc. IEEE Veh. Technol. Conf.*, Sep. 2012, pp. 1–5.
[43] A. Merwaday and I. Guvenc, "UAV assisted heterogeneous networks for public safety communications," in *Proc. IEEE Wireless Commun. Netw. Conf. Workshops (WCNCW)*, Mar. 2015, pp. 329–334.
[44] H. C. Nguyen, R. Amorim, J. Wigard, I. Z. Kovács, T. B. Sørensen, and P. E. Mogensen, "How to ensure reliable connectivity for aerial vehicles over cellular networks," *IEEE Access*, vol. 6, pp. 12304–12317, 2018.
[45] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: Opportunities and challenges," *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 36–42, May 2016.
[46] Q. Zhang, M. Jiang, Z. Feng, W. Li, W. Zhang, and M. Pan, "IoT enabled UAV: Network architecture and routing algorithm," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3727–3742, Apr. 2019.
[47] A. Orsino *et al.*, "Effects of heterogeneous mobility on D2D- and drone-assisted mission-critical MTC in 5G," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 79–87, Feb. 2017.
[48] S. Qazi, A. S. Siddiqui, and A. I. Wagan, "UAV based real time video surveillance over 4G LTE," in *Proc. Int. Conf. Open Source Syst. Technol. (ICOSST)*, Dec. 2016, pp. 141–145.

[49] N. H. Motlagh, T. Taleb, and O. Arouk, "Low-altitude unmanned aerial vehicles-based Internet of Things services: Comprehensive survey and future perspectives," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 899–922, Dec. 2016.
[50] C. Ting, X. Yun, Z. Xiangmo, G. Tao, and X. Zhigang, "4G UAV communication system and hovering height optimization for public safety," in *Proc. IEEE 19th Int. Conf. E-Health Netw., Appl. Services (Healthcom)*, Dalian, China, Oct. 2017, pp. 1–6.
[51] P. Chandhar, D. Danev, and E. Larsson, "Massive MIMO for communications with drone swarms," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 1604–1629, Mar. 2018.
[52] S. Hayat, E. Yanmaz, and R. Muzaffar, "Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2624–2661, 4th Quart., 2016.
[53] I. Bor-Yaliniz and H. Yanikomeroglu, "The new frontier in RAN heterogeneity: Multi-tier drone-cells," *IEEE Commun. Mag.*, vol. 54, no. 11, pp. 48–55, Nov. 2016.
[54] F. Cheng *et al.*, "UAV trajectory optimization for data offloading at the edge of multiple cells," *IEEE Trans. Veh. Technol.*, vol. 67, no. 7, pp. 6732–6736, Jul. 2018.
[55] H. Simaremare, A. Syarif, A. Abouaissa, R. F. Sari, and P. Lorenz, "Performance comparison of modified AODV in reference point group mobility and random waypoint mobility models," *IEEE Int. Conf. Commun.*, Jun. 2013, pp. 3542–3546.
[56] S. Rosati, K. Krużelecki, L. Traynard, and B. Rimoldi, "Speed-aware routing for UAV ad-hoc networks," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2013, pp. 1367–1373.
[57] J.-D. M. Me Biomo, T. Kunz, and M. St-Hilaire, "Routing in unmanned aerial ad hoc networks: Introducing a route reliability criterion," in *Proc. 7th IFIP Wireless Mobile Netw. Conf. (WMNC)*, May 2014, pp. 1–7.
[58] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Drone small cells in the clouds: Design, deployment and performance analysis," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2015, pp. 1–6.
[59] S. Koulali, E. Sabir, T. Taleb, and M. Azizi, "A green strategic activity scheduling for UAV networks: A sub-modular game perspective," *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 58–64, May 2016.
[60] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Optimal transport theory for power-efficient deployment of unmanned aerial vehicles," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2016, pp. 1–6.
[61] C. Zhang and W. Zhang, "Spectrum sharing for drone networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 1, pp. 136–144, Jan. 2017.
[62] G. Gankhuyag, A. P. Shrestha, and S.-J. Yoo, "Robust and reliable predictive routing strategy for flying ad-hoc networks," *IEEE Access*, vol. 5, pp. 643–654, 2017.
[63] M. Alzenad, A. El-Keyi, F. Lagum, and H. Yanikomeroglu, "3-D placement of an unmanned aerial vehicle base station (UAV-BS) for energy-efficient maximal coverage," *IEEE Wireless Commun. Lett.*, vol. 6, no. 4, pp. 434–437, Aug. 2017.
[64] N. Zhao *et al.*, "Caching UAV assisted secure transmission in hyper-dense networks based on interference alignment," *IEEE Trans. Commun.*, vol. 66, no. 5, pp. 2281–2294, May 2018.
[65] J. Jiang and G. Han, "Routing protocols for unmanned aerial vehicles," *IEEE Commun. Mag.*, vol. 56, no. 1, pp. 58–63, Jan. 2018.
[66] C.-M. Cheng, P.-H. Hsiao, H. T. Kung, and D. Vlah, "Maximizing throughput of UAV-relaying networks with the load-carry-and-deliver paradigm," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2007, pp. 4420–4427.
[67] M. Wzorek, D. Landen, and P. Doherty, "GSM technology as a communication media for an autonomous unmanned aerial vehicle," in *Proc. 21st Bristol UAV Syst. Conf.*, Apr. 2006.
[68] C. M. Joslin, E. R. Martinez, C. N. Saikali, C. A. Tran, and P. P. Burke, "Drone mapping of radio frequency signals in 3D," Senior Thesis, UC Irvine, Irvine, CA, USA, 2017, pp. 1–22.
[69] Qualcomm. (2017). *LTE Unmanned Aircraft Systems*. [Online]. Available: https://www.qualcomm.com/documents/lte-unmanned-aircraft-systems-trial-report
[70] B. Van Der Bergh, A. Chiumento, and S. Pollin, "LTE in the sky: Trading off propagation benefits with interference costs for aerial nodes," *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 44–50, May 2016.
[71] *Make Your Personal Drone Fly Even Farther With a 4G Network Connection*. Accessed: Apr. 5, 2019. [Online]. Available: https://wiredcraft.com/blog/drone-copter-uav-4g-network
[72] X. Lin *et al.*, "The sky is not the limit: LTE for unmanned aerial vehicles," *IEEE Commun. Mag.*, vol. 56, no. 4, pp. 204–210, Apr. 2018.
[73] R. Amorim, H. Nguyen, P. Mogensen, I. Z. Kovács, J. Wigard, and T. B. Sørensen, "Radio channel modeling for UAV communication over cellular networks," *IEEE Wireless Commun. Lett.*, vol. 6, no. 4, pp. 514–517, Aug. 2017.
[74] A. Al-Hourani and K. Gomez, "Modeling cellular-to-UAV path-loss for suburban environments," *IEEE Wireless Commun. Lett.*, vol. 7, no. 1, pp. 82–85, Feb. 2018.
[75] E. Teng, J. D. Falcão, and B. Iannucci, "Holes-in-the-Sky: A field study on cellular-connected UAS," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Jun. 2017, pp. 1165–1174.
[76] F. Maiwald and A. Schulte, "Using LTE-networks for UAS-communication," in *Proc. 36th Eur. Telemetry Test Conf.*, 2016, pp. 166–175.
[77] Z. Yuan, J. Jin, L. Sun, K.-W. Chin, and G.-M. Muntean, "Ultra-reliable IoT communications with UAVs: A swarm use case," *IEEE Commun. Mag.*, vol. 56, no. 12, pp. 90–96, Dec. 2018.
[78] Y. Zhang and Z. Yuan, "Cloud-based UAV data delivery over 4G network," in *Proc. 10th Int. Conf. Mobile Comput. Ubiquitous Netw. (ICMU)*, Oct. 2017, pp. 1–2.
[79] (2016). *Press Release: Ericsson and China Mobile Conduct World's First 5G Drone Prototype Field Trial*. Accessed: Apr. 6, 2019. [Online]. Available: https://www.ericsson.com/en/news/2016/8/ericsson-and-china-mobile-conduct-worlds-first-5g-drone-prototype-field-trial-
[80] (2017). *Press Release: KDDI and Terra Drone Have Announced Completion of Inventing '4G LTE Control System'*. Accessed: Apr. 6, 2019. [Online]. Available: https://www.terra-drone.net/en/kddi-and-terra-drone-have-announced-completion-of-inventing-4g-lte-control-system/
[81] *GSMA Internet of Things Case Study How Cellular Technology Enables Anti-Fire Drones*. Accessed: Apr. 5, 2019. [Online]. Available: https://www.gsma.com/iot/wp-content/uploads/2019/02/GSMA-Anti-fire-Drones-Case-Study.pdf
[82] FAA. *Civil Airmen Statistics*. Accessed: Apr. 11, 2019. [Online]. Available: https://www.faa.gov/data_research/aviation_data_statistics/civil_airmen_statistics/
[83] USDOT. Accessed: Apr. 11, 2019. *FAA Drone Registry Tops One Million*. [Online]. Available: https://www.transportation.gov/briefing-room/faa-drone-registry-tops-one-million
[84] R. A. Clothier, B. P. Williams, and K. J. Hayhurst, "Modelling the risks remotely piloted aircraft pose to people on the ground," *Saf. Sci.*, vol. 101, pp. 33–47, Jan. 2018.
[85] A. Washington, R. A. Clothier, and J. Silva, "A review of unmanned aircraft system ground risk models," *Prog. Aerosp. Sci.*, vol. 95, pp. 24–44, Nov. 2017.
[86] D. R. Arterburn, C. T. Duling, and N. R. Goli, "Ground collision severity standards for UAS operating in the national airspace system (NAS)," in *Proc. 17th AIAA Aviat. Technol. Integr. Oper. Conf.*, Jun. 2017, pp. 1–16.
[87] P. Gregg. *Risk in the Sky?*. Accessed: Apr. 9, 2019. [Online]. Available: https://udayton.edu/blogs/udri/18-09-13-risk-in-the-sky.php
[88] B. M. Schulman, *DJI Demands Withdrawal of Misleading Drone Collision Video*. Accessed: Apr. 9, 2019. https://www.dji.com/newsroom/news/dji-demands-withdrawal-of-misleading-drone-collision-video
[89] B. Simpson. *The Science of Crashing into Stuff (Kinetic Energy)*. Accessed: Apr. 9, 2019. [Online]. Available: https://www.youtube.com/watch?v=9RDnWaDjV9E&t=655s
[90] B. Simpson. *Will This Hurt?*. Accessed: Apr. 9, 2019. [Online]. Available: https://www.youtube.com/watch?v=-1XMvn7ddyw
[91] *ASSURE Annual Report 2018*. Accessed: Apr. 11, 2019. [Online]. Available: http://www.assureuas.org/annual-reports/ASSURE_2018AR.pdf
[92] D. Arterburn, M. Ewing, R. Prabhu, F. Zhu, and D. Francis. *UAS Ground Collision Severity Evaluation*. Accessed: Apr. 11, 2019. [Online]. Available: www.assureuas.org/projects/deliverables/sUASGroundCollisionReport.php
[93] J. Poss. *Was There an Oops in the Proposed OPS Over People Rules?*. Accessed: Apr. 10, 2019. [Online]. Available: http://insideunmannedsystems.com/was-there-an-oops-in-the-proposed-ops-over-people-rules/
[94] FAA. *Operation of Small Unmanned Aircraft Systems Over People*. Accessed: Apr. 11, 2019. [Online]. Available: https://www.regulations.gov/searchResults?rpp=25&po=0&s=FAA-2018-1087&fp=true&ns=true

[95] W. Sneader, "The discovery of aspirin: A reappraisal," *BMJ*, vol. 321, no. 7276, pp. 1591–1594, 2002.
[96] M. Podhradsky, C. Coopmans, and N. Hoffer, "Improving communication security of open source UAVs: Encrypting radio control link," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Jun. 2017, pp. 1153–1159.
[97] C. Bunse and S. Plotz, "Security analysis of drone communication protocols," in *Engineering Secure Software and Systems*. Cham, Switzerland: Springer, 2018, pp. 96–107.
[98] NAE. *FAA Should Change Its Safety Risk Assessment Approach for Drones to Effectively Integrate Them Into the Nation's Airspace*. Accessed: Apr. 9, 2019. [Online]. Available: http://www8.nationalacademies.org/onpinews/newsitem.aspx?RecordID=25143
[99] *Assessing the Risks of Integrating Unmanned Aircraft Systems (UAS) into the National Airspace System*, Nat. Academies Sci. Eng. Med., Washington, DC, USA, 2018.
[100] *Telemetry and HD video over 4G*. Accessed: Apr. 5, 2019. [Online]. Available: https://diydrones.com/forum/topics/telemetry-and-hd-video-over-4g
[101] L. Torvalds, "Linux: A portable operating system," M.S. thesis, Dept. Comput. Sci., Univ. Helsinki, Helsinki, Finland, 1997.
[102] N. H. Motlagh, M. Bagaa, and T. Taleb, "UAV-based IoT platform: A crowd surveillance use case," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 128–134, 2017.
[103] G. E. Moore, "The experts look ahead. Cramming more components onto integrated circuits," *Electronics*, vol. 38, no. 8, pp. 114–119, 1965.
[104] *Skydrone*. Accessed: Apr. 9, 2019. [Online]. Available: https://www.skydrone.aero/
[105] GlobalUAV. *Global UAV Technologies*. Accessed: Apr. 9, 2019. [Online]. Available: https://globaluavtech.com/
[106] *Flytbase*. Accessed: Apr. 9, 2019. [Online]. Available: https://flytbase.com/
[107] Cape. *Cape Aerial Telepresence*. Accessed: Apr. 9, 2019. [Online]. Available: https://www.cape.com/
[108] *Flythere*. Accessed: Apr. 9, 2019. [Online]. Available: https://www.flythere.com/
[109] *Zerotier*. Accessed: Apr. 9, 2019. [Online]. Available: https://www.zerotier.com/
[110] *UAVMatrix*. Accessed: Apr. 9, 2019. [Online]. Available: https://uavmatrix.com/
[111] *4G/LTE Softmod for the Parrot Disco*. Accessed: Apr. 12, 2019. [Online]. Available: https://github.com/uavpal/disco4g
[112] *Parrot Disco Specifications*. Accessed: Apr. 12, 2019. [Online]. Available: https://www.parrot.com/us/drones/parrot-disco-fpv#-parrot-disco-fpv
[113] *Black Hornet Data Sheet*. Accessed: Apr. 10, 2019. [Online]. Available: https://www.flir.com/globalassets/imported-assets/document/black-hornet-prs-spec-sheet.pdf
[114] *FLIR to Deliver Black Hornet PRS UAV for US Army SBS Programme SHARE*. Accessed: Apr. 9, 2019. [Online]. Available: https://www.army-technology.com/news/flir-black-hornet-sbs-programme/
[115] *US Air Force MQ-9 Reaper Fact Sheet*. Accessed: Apr. 10, 2019. [Online]. Available: https://www.af.mil/About-Us/Fact-Sheets/Display/Article/104470/mq-9-reaper/
[116] M. Hassanalian and A. Abdelkefi, "Classifications, applications, and design challenges of drones: A review," *Prog. Aerosp. Sci.*, vol. 91, pp. 99–131, May 2017.
[117] A. Mairaj, A. I. Baba, and A. Y. Javaid, "Application specific drone simulators: Recent advances and challenges," *Simul. Model. Pract. Theory*, vol. 94, pp. 100–117, Jan. 2019.
[118] *Academy of Model Aeronautics*. Accessed: Apr. 9, 2019. [Online]. Available: https://www.modelaircraft.org/
[119] D. McCullough, *The Wright Brothers*. Simon and Schuster, New York, NY, USA, 2015.
[120] J. Lyu, Y. Zeng, and R. Zhang, "UAV-aided offloading for cellular hotspot," *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 3988–4001, Jun. 2018.
[121] USAF. *Multifunctional Integrated Sensing Cargo Pocket UAS*. Accessed: Apr. 9, 2019. [Online]. Available: https://www.sbir.gov/sbirsearch/detail/1532159
[122] D. Wu *et al.*, "ADDSEN: Adaptive data processing and dissemination for drone swarms in urban sensing," *IEEE Trans. Comput.*, vol. 66, no. 2, pp. 183–198, Feb. 2017.

PETER J. BURKE (M'02–SM'17) received the Ph.D. degree in physics from Yale University, New Haven, CT, USA, in 1998. From 1998 to 2001, he was a Sherman Fairchild Postdoctoral Scholar in physics with the California Institute of Technology, Pasadena, CA, USA. Since 2001, he has been a Faculty Member with the Department of Electrical Engineering and Computer Science, University of California at Irvine, Irvine, CA, USA. His current research interests include EECS, BME, chemical and biomolecular engineering, materials science and engineering, and chemical and materials physics.