SSC16–IX–05

Supporting the Flock: Building a Ground Station Network for Autonomy and Reliability

Kyle Colton, Bryan Klofas
Planet Labs
346 9th Street, San Francisco CA 94103; 858–304–0302
kyle.colton@planet.com

ABSTRACT

Planet Labs is on a mission to image the whole earth every day. This requires a large orbital constellation and a distributed, autonomous ground station network. The network currently includes 11 geographically diverse sites, many with multiple antenna systems. Antennas systems are deployed with uniform equipment to simplify interfaces and ease operations. Additionally, tools have been developed for automated monitoring and remote troubleshooting of RF chains. Predictive modeling tools help to plan for future need as the orbital constellation grows. Lessons learned and ideas for automation and monitoring are also discussed.

A NEW IMAGE OF THE EARTH EVERY DAY

Planet Labs is on a mission to image the whole world every day and make global change visible, accessible, and actionable. We strive to improve the world by providing a medium-resolution rapidly-updating data set. Recently our data has been used to raise awareness about illegal mining in a natural reserve[1]. In response to the recent fires in Alberta [2] and earthquake in Ecuador [3], we released our imagery of the areas to aid government and humanitarian relief efforts.

As of May 2016, we have successfully launched 133 Dove satellites from 11 different launch vehicles and the International Space Station, and we are rapidly scaling operations to meet our mission. Our current constellation consists of fifty-five 3U CubeSats. Planet has built the largest constellation of commercial earth imaging satellites, and our Doves image anytime they are over land. This constellation generates massive amounts of image data that we need to bring to the ground in a quick and efficient manner. In addition, we collect normal housekeeping telemetry, update software, and refresh the on-board schedules.

MEETING THE NEEDS OF AGILE AEROSPACE

Agile aerospace is a new set of principles based on the agile software development concepts including rapid design iteration, a continuous improvement process, and easy scalability. As a result of Planet's agile approach, recent ISS deployments have dramatically increased our constellation of satellites (or "Flock of Doves") to fifty-five on-orbit CubeSats. Our Doves take approximately 650 passes per day on 33 antenna systems spread across our 11 active ground station sites. The average amount of picture data downlinked per day is approximately 550 GB, with a maximum of 777 GB [4]. Once these latest satellites finish commissioning and begin operations, downlink numbers will rise further.

Our distributed ground station network provides several benefits; it reduces the impact of any one failure, serves multiple orbital planes, reduces the time between image capture and downlink, and spreads the load of satellite downlinks more evenly across the globe. At the same time, the multiple sites increase complexity of the network and result in a global, "always-on" ground station network that is maintained, monitored, developed, and expanded by a relatively small team of five people.

Autonomy and reliability are necessary. Passes and tasks are scheduled by a computer model, faults are detected and cleared as much as possible by the individual stations, and those not automatically resolved are escalated to the on-call engineer. We rely on software tools to aid in managing a network capable of supporting the current and future constellation.

Building Uniform Ground Stations

The first Dove CubeSats were launched in April 2013, and there have been 14 satellite hardware revisions since then that have included new processors, better persistent data storage, newer cameras, and more advanced optics. Even with these hardware changes the link interface has been consistent throughout this time period. Aside from better antennas and a recent baud rate change (from 24 Mbaud to 60 Mbaud) that required a hardware upgrade, all of the changes have been software.

On the ground station side of the link, there have been several hardware upgrades and countless software updates during the past three years. Hardware changes include new internet firewall appliances and ethernet switches for increased network security, new transceivers with reduced EMI and spurious signals, GPS time synchronization, and new computer servers with more processing power and remote management features.

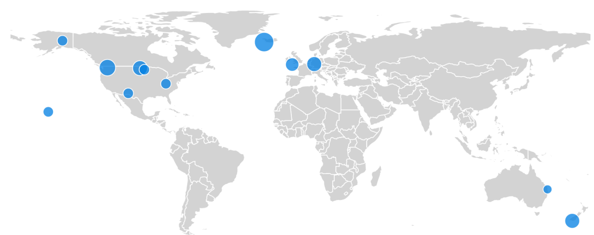

Figure 1: Planet's Ground Station Network.

Our ground station network includes 11 active global sites. While Figure 1 shows that most of our ground station sites are located in the United States, we do have sites scattered around the world. As discussed previously [5, 6], most CubeSats are using amateur radio frequencies, which are licensed for noncommercial use worldwide. Since we are a commercial company based in the United States, Planet acquired commercial licenses for our satellites [7]. Working with the different frequency licensing organizations around the world can be challenging and time consuming.

Uniformity, however, aids in licensing. Licenses often require knowledge of the hardware, and by keeping it similar from site to site, we can begin the licensing exploration and application process in advance.

Tracking, Telemetry, and Command

The Dove satellites use UHF for TT&C and orbit determination [9], so the constellation benefits greatly from having frequent and global UHF link opportunities. The Planet Labs UHF ground station uses simple, inexpensive COTS components to support the need for diverse coverage at a reasonable cost and with easy maintenance. They use standard Yagi antennas positioned by a Yaesu rotor for both uplink and downlink. Figure 2 shows the antennas on a typical UHF ground station.

Figure 2: Example UHF Station.

The only non-COTS component in our ground stations is the custom-built SpaceTalker transceiver. It is based on a Texas Instruments CC1110 wireless MCU, which has both a UHF transceiver and microcontroller unit in a single chip. The server communicates with the transceiver over USB [8]. Both the ground SpaceTalker radio and space hardware use extremely similar architecture and interfaces, increasing the system's predictability and reliability. Figure 3 shows a block diagram of our UHF systems.

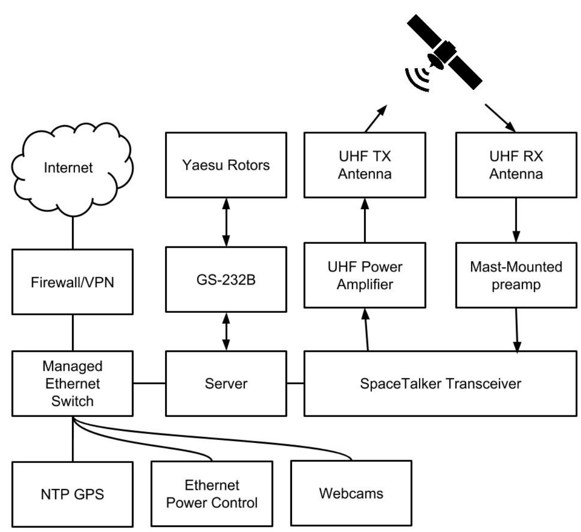

Figure 3: UHF TT&C Block Diagram.

High-Speed Downlink

In addition to the UHF capabilities of the Planet Labs SpaceTalker radio, there is another CC1110 chip with upconversion to S-band. This is our high-speed uplink radio for uploading commands and new software to the constellation. We use 8.2 GHz for our high-speed data downlink. The encoding scheme is DVB-S2, which allows for different modulations, forward-error correction settings.

While developed for satellite television applications, the DVB-S2 standard includes a Generic Stream Encapsulation (GSE) protocol which allows any digital data to be transmitted within DVB-S2 frames. Within the GSE, we use Internet Protocol formatting. Combined with the on-board processor, which runs vanilla Ubuntu, this formatting allows us to securely connect with our satellites and run commands, scripts, or anything that can be done on a normal computer on the ground. During nominal operations, the encrypted link is used for downlinking pictures and logs.

On the ground station side, we have three types of dishes spread amongst our six S/X-band sites. All of our dishes are designed to 29 dB/K or better. Figure 4 shows a block diagram of our high-speed downlink stations. Note the similarity with our UHF TT&C stations.

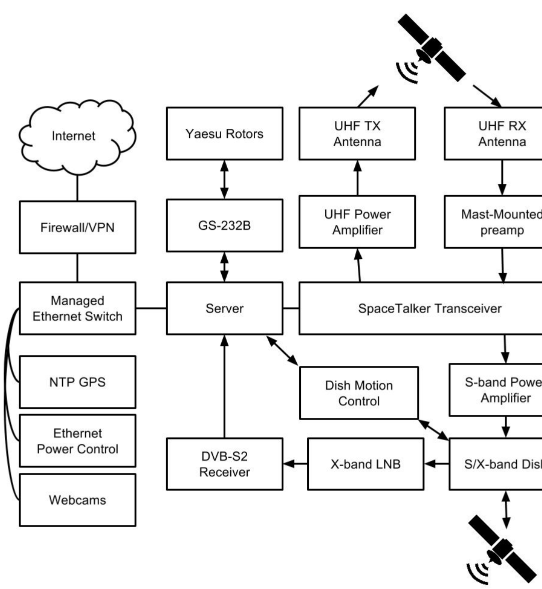

Figure 4: High-Speed Downlink Block Diagram.

Automating the Downlink

At a fundamental level, our satellites have a very simple ConOps: when over land, point down and take pictures. Occasionally, the Doves take pictures of the moon for optical calibration, desaturate wheels, or turn on the star camera for attitude calibration. When a downlink session is scheduled, the Dove turns on the X-band transmitter and tracks the ground station as it passes under the satellite. Whenever the satellites are not doing the above, they charge batteries in a low-drag state.

This basic ConOps cries out for a fully automated system; Planet Labs has fifty-five satellites in orbit and will be scaling rapidly in the next year. Planet Labs planned for satellite and ground station automation from the beginning in order to allow the Flock to scale as necessary. Our operations team consists of five “spaceship captains” responsible for commissioning new satellites, anomaly resolution on existing satellites, and building tools to improve on-orbit operations. A high degree of autonomy is necessary for such a small team to support the on-orbit constellation.

These same automation concepts are applied to the ground station network. The ground station team, which also consists of five people, does not have the time nor the ability to manually control each of the ground stations during its downlink sessions. Just before each downlink session, the ground station downloads the pass schedule, satellite commands (or tasks) to run during the pass, and new orbital elements from Mission Control for antenna pointing.

After the pass, pictures are uploaded to Amazon Web Services (AWS), a cloud-computing platform. Satellite and ground station telemetry and pass logs are sent back to Mission Control. The Mission Control website, also hosted on AWS, provides the Spaceship Captains and ground station engineers up-to-date information on each satellite and ground station. Statistics generated for each pass include tasks run, gigabytes downloaded, commands uploaded, satellite health, and ground station status.

Mission Control is written mostly in Python, and schedules approximately 350 passes per day on the UHF TT&C systems and 300 X-band passes per day. Mission Control is built by an in-house team of three software engineers. We found that none of the existing products fit our needs, and most were closed-source products. As we scale up our constellation to hundreds of satellites in space, open access to the source code is necessary for new features and bug fixes.

FAULT DETECTION, NOTIFICATION, AND CLEARING

Planet's ground stations use many open source packages for fault detection. Often, these software packages are designed for IT professionals for website and server monitoring but can be applied to other scalable networks, such as ground stations.

Much of the fault detection and notification is built around an open source network monitoring service. This service runs on the ground station servers and continually monitors the health and status of the hardware and software. When a fault is detected, the local machine notifies the cloud-based central server, which then alerts our internal messaging system and an external incident notification service. The service is configured to notify the "on-call" engineer, handle acknowledgements, and escalate alerts when needed.

While the ground station team strives for 100 percent uptime, at any given time some ground stations can be out of service due to equipment failure. It is much less expensive to build for 90% uptime versus 100% uptime, so the network is purposely slightly under capacity, and therefore more fault-tolerant. Satellite tasks are also robust to a single station failure; Mission Control tracks acknowledged commands and will automatically re-run incomplete or unacknowledged commands during the next pass. In addition, each failure gives us new insight into our next upgrades to improve the network.

If a fault requires hardware to be adjusted or fixed, we do have "remote hands" contractors for each site that are able to work with us to diagnose and fix issues from afar.

In the event of power-loss to a site, the hardware is all designed to come back up automatically in a known good state and resume activities by polling Mission Control and rescheduling the next passes.

REMOTE TROUBLESHOOTING

While it is easiest to to address an issue directly onsite, visits take time. We have built tools to reduce that time and personnel cost by solving issues from afar. This both allows us to clear some issues immediately and to provide instruction to our local help or plan for when we do need to visit the site.

One of our most basic tools is a network-based power switch. From the switch, we can power-cycle components remotely and view current consumption. It is instrumental for testing the system in various configurations. For example, a quick power cycle of the LNA in the system can tell us if it is still operating as expected. If there is no appreciable difference in system performance, we can quickly narrow in on investigating a fault before or at the LNA.

The Planet Labs open source radio also gives us invaluable tools in troubleshooting [10]. At the most basic level, we can poll the RF power received by the TI CC1110 receiver. Combined with our ability to remotely power parts of the receive and transmit chains, we can quickly test the functionality of individual components.

Using the RF power at the receiver, we can rotate antennas and collect measurements at all azimuth angles to get a noise-survey of a site. These surveys can give us quick insights to how an antenna is performing. A normal site, as seen in Figure 5, will have very uniform plots with a bit of variation in noise as the antenna rotates. We expect variation at the lowest elevations, so absence of noise can be just as useful an indicator as too much noise in any given direction.

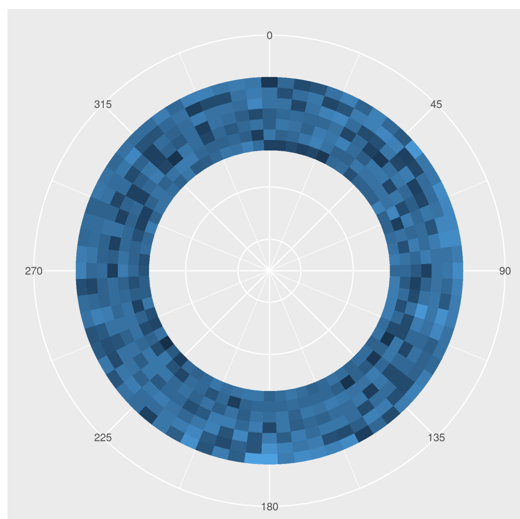

Figure 5: Example Noise Survey.

We have used these tools to identify local sources of interference that have hampered our link performance, and as an additional troubleshooting option in the transmit chain. For example, at one site an amplifier was continuously biased. The problem manifested itself as increased noise at the site, impacting operations. Figure 6 shows surveys from other antennas at the site that we used to triangulate the source of the issue and subsequently fix it.

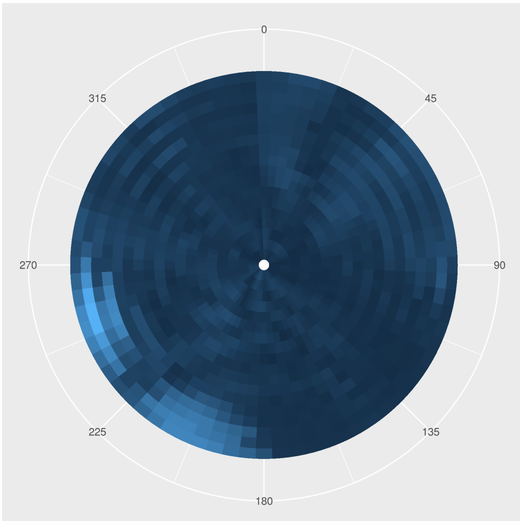

Figure 6: Faulty Amplifier Noise Survey

The noise survey program we wrote moves clockwise in azimuth before stepping in elevation and returning counter-clockwise in azimuth. Figure 7 shows a marked difference between the noise seen in clockwise motion and counterclockwise motion. Reversing the direction of the program changed the pattern to reflect the new clockwise and counterclockwise, enabling us to point our local contractor to an internally-separated coaxial cable.

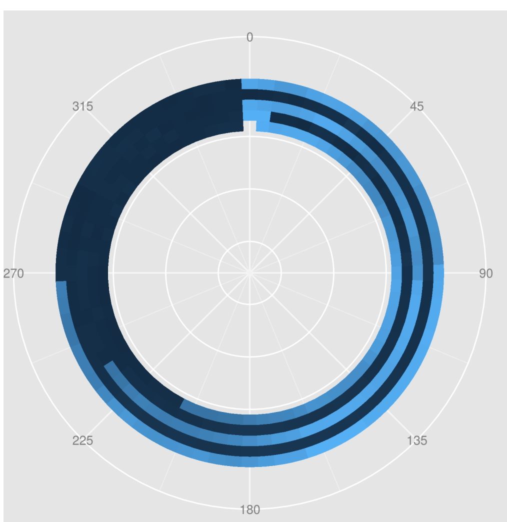

Figure 7: Motion-Dependent Fault.

Tracking with Multiple Systems

Tracking the same satellite with multiple systems at a single site is another very useful troubleshooting method. We can collect and store the receive power on our X-band systems throughout a pass, then overlay them at the end to get a rough comparison of the two systems. This also helps to differentiate between what anomalies are satellite specific, and what are ground station specific. Co-tracking can also highlight minor errors in pointing; one of the systems may show a drop-off at the Time of Closest Approach (TCA), when the dish is required to move the fastest to keep up with the LEO satellite, and when precise pointing is most important.

The red and purple lines in Figure 8 show Carrier-to-noise-plus-interference ratios (CNI) on two nearby dishes tracking the same satellite. The dip in the red dish shows that it was not pointing as accurately at the satellite during the middle of the pass. Link margins for these same two dishes are shown with the green and blue lines. The step changes at the beginning and end of the pass are due to MODCOD/data rate changes.

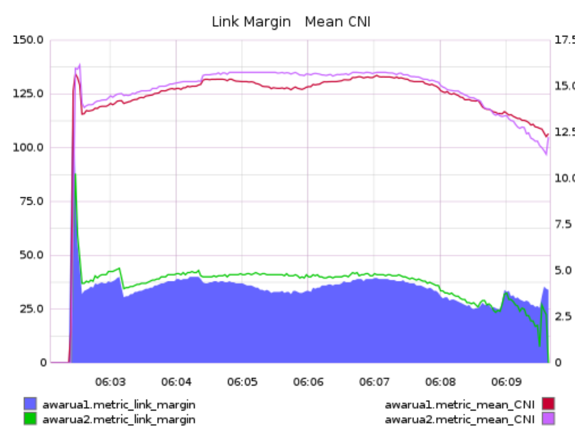

Figure 8: Cotracking satellites.

Additionally with co-tracking, we can transmit commands up on one system and receive them via another system at the same site. This method verifies the entire link chain, and when combined with receive power readings from the spacecraft and the ground segment, can provide valuable information about both the receive and transmit chains of each system. Lastly, if any portion of the chain is not functioning, it quickly becomes evident by co-tracking with a known good site.

Tracking Known Objects

When nothing else is available, or when a system is still in commissioning and not ready for active downlink, we can always fall back to using a known object, such as the sun, the moon, or one of the well-documented earth observation satellites such as Terra or Aqua.

In addition to monitoring RF performance using receivers, we can monitor antenna movement, replay past tracks, and check motion using remote cameras at each dish. The cameras are an easy way to get a set of eyes on the dish and equipment from afar and can watch even when people cannot. The cameras give an independent check of hardware status and have helped us with equipment errors more than once. We can try to be robust to failure, but troubleshooting and recovery methods are invaluable for getting operational quickly after an unforeseen problem. Being agile on the ground is not about avoiding mistakes, rather it is about being able to quickly learn about problems, gracefully recover, and institute procedures to prevent future problems.

Building Tools for Troubleshooting

Since much of the equipment and software is the same across the network, we have built very standardized testing tools to automate tasks. Software is kept in centrally controlled repositories and can be redeployed to any or all stations automatically, leaving us with a high degree of confidence about how each system will perform and about what tools we can use at a site.

As much as possible, we design tools to be versatile and accept different configurations. Much of this is done through abstraction layers. For example, interface code may reside in a library and be called upon by a service. Troubleshooting tools then call upon the same library to run. By reusing production code in troubleshooting, we stay as close to the original environment as possible. Additionally, since we use unix-like operating systems as our primary development, work, and production environments, we can use many of the standard shell commands and native tools to pipe outputs between our scripts or to expand upon the normal scope of the programs.

PREDICTIVE MODELING OF FUTURE NEED

Agile aerospace moves quickly; in an iterate-launch cycle, the longest lead time is often the launch. However, licensing and building ground stations can be long-lead items. Simulations inform us on how much capacity to build and prevent us from over-building, a waste of both time and capital.

Orbit propagation tools generate both an estimate of the total number of Doves needed for our high cadence, whole-earth imagery, as well as the estimated pass lengths over any given latitude. By knowing the total number of satellites needed and the data generated by the Doves, we can estimate the necessary total contact time the ground station network must supply for any give location. New ground station locations are then optimized for cost, data capacity, and location.

By understanding the operating cost with respect to the data downlink capacity of each station, we can quickly generate a comparison cost to those provided by third party vendors. This calculation feeds back into decisions regarding purchasing excess capacity either in the case of a site failure or simply during high-use time periods, like when commissioning new satellites.

LESSONS LEARNED

Agile aerospace is a focus on iterative design to continuously upgrade a product. While much of the hardware can be expensive and we do not have an "agile" dish, we can iterate around the expensive hardware and support it with an agile ground build. Control software and systems can improve over time without necessitating change of the larger hardware components. Additionally, external components can be added to improve an already existing system. For example, we are currently deploying GPS time sources to many of our sites. Existing NTP time sources gives us adequate timing, but an upgrade to local stratum 1 time sources allows better orbit determination, link encryption, and positional knowledge onboard the satellites. Time sources are just one example of ground systems that we can iterate and improve upon to make the system better as a whole.

Additionally, it is very helpful to keep a lab setup of as much of the ground hardware as feasible. Not only does this provide a better testing platform for the development of both the ground and the space systems in the lab, it also provides fast access to a system for investigating how things may have failed on a site. The lab is a far easier build and test environment than the field, so having the lab setup can also provide information about how to streamline field installations.

While we try to be agile, licensing and the physical build can often take much longer than expected. We have found it very important to begin the licensing and contracting processes well before there is any need for the site. For that reason, modeling the future system requirements can be invaluable.

SUPPORTING AGILE AEROSPACE FROM THE GROUND

By focusing on design iteration, flexibility, and rapid change we can emulate agile aerospace. Not every aspect of a ground station can be agile, but by keeping those principles in mind, we can implement agile aerospace in other aspects of the network architecture. Software development and deployment can be centrally controlled and deployed to allow improvements as the system develops. Capacity can be modeled to plan coverage expansion as needed. Remote troubleshooting tools can allow a central team to control all aspects of a station from afar and manage any issues that arise.

Planet Labs' ground station network is built with the idea of easy expansion and extensibility. At every point, we reflect on what we can change to improve capabilities while sustaining the system as a whole. The Planet Labs ground station network works daily to support the on-orbit Flock and is ready to grow and advance in parallel with our satellites as we scale to a new image of the earth every day.

REFERENCES

[1] *Planet Pulse*. https://www.planet.com/pulse/illegal-gold-mine-encroaches-on-protected-rainforest/.

[2] *Planet Pulse*. https://www.planet.com/pulse/fort-mcmurray-wildfire/.

[3] *Planet Pulse*. https://www.planet.com/pulse/ecuador-earthquake-data-available-under-open-license/.

[4] Klofas, Bryan. "Planet Labs Ground Station Network." Presented at the 13th Annual CubeSat Developers Workshop. San Luis Obispo, CA. April 2016.

[5] Matas, Attila. "ITU Radio Regulations Related to Small Satellites." Presented at the 10th Annual CubeSat Developers Workshop. San Luis Obispo, CA. April 2013.

[6] Persaud, Sankar. "Regulatory matters concerning authorization of Small Sats: Common Misunderstandings about the FCC and Licensing." Presented at the 2012 CubeSat Summer Workshop. Logan, Utah. August 2012.

[7] FCC. "Guidance On Obtaining Licenses For Small Satellites." Federal Communications Commission Public Notice DA-13-445. March 2013.

[8] Hallam, Henry. "Design and Results of the Planet Labs Low-Speed Transceiver Radio." Presented at the 2014 CubeSat Summer Workshop. Logan, Utah. August 2014.

[9] Foster, Cyrus, Henry Hallam, James Mason. "Orbit Determination and Differential-Drag Control of Planet Labs CubeSat Constellations." `arXiv:1509.03270v1`. AAS 15-524.

[10] Hallam, Henry, Matt Ligon, Kiruthika Devaraj, Alex Ray, Ryan Kingsbury. "Open Source Low-Speed Transceiver Solution for CubeSats." Proceedings of the 13th Annual Summer CubeSat Developers' Workshop. Logan, Utah. August 2016.