Haystack GSI:

A Satellite Ground Station Interface Built with GNURadio

Michael Gutierrez[1], **Maximilian Riccioli**[2], T. Lucas Briggs[3], Tobias Gedenk[4], Frank Lind[4], Mary Knapp[4], John Swoboda[4], Ryan Volz[4]

[1] California Institute of Technology, [2] University of Texas at Austin, [3] Northeastern University, [4] MIT Haystack Observatory,

MIT HAYSTACK OBSERVATORY

GRCon 22 WASHINGTON, D.C.

AERO/VISTA Introduction

Ground Operations

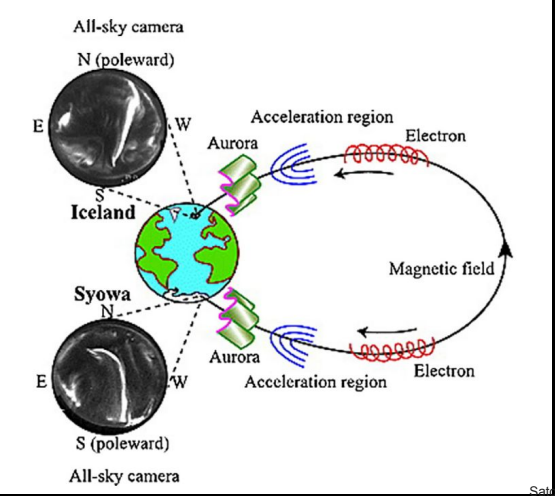
Haystack GSI Overview

Routing Data with GNURadio

Future Development

Earth's aurora: optical signature

- Electrons channeled into the Earth's atmosphere by strong magnetic fields create electromagnetic disturbances
- Optical emissions at around 150km, caused by electrons colliding with neutral atmospheric particles

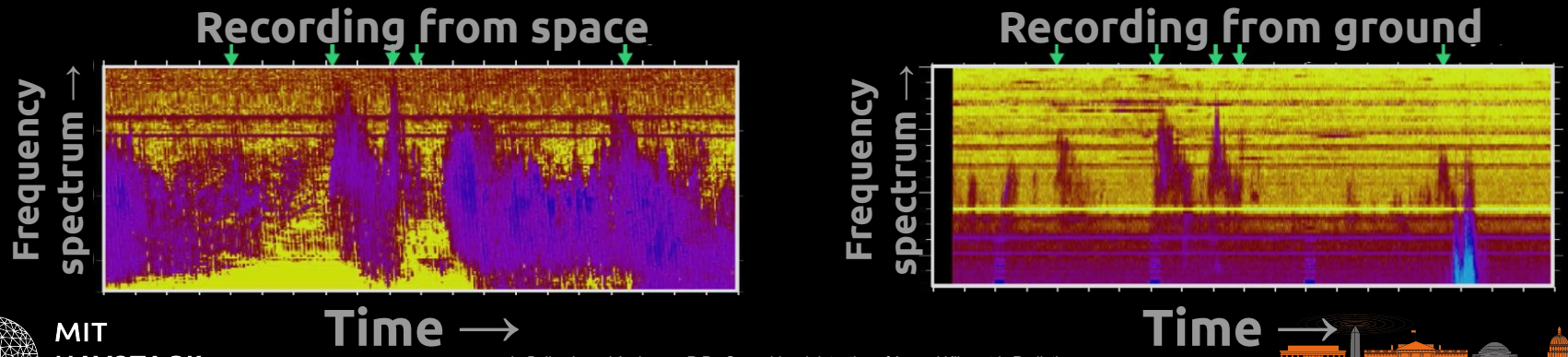


Sato, N., Kadokura, A., Ejiri, M., Deguchi, H. 2005. Tracing geomagnetic conjugate points using exceptionally similar synchronous auroras. Geophys. Res. Lett. 32, 17.

Unanswered questions about Earth's radio aurora

- Radio emissions occur above 500km
- From plasma theory & radio propagation, we don't expect to see radio disturbances <5MHz through the low-frequency-opaque atmosphere… but sometimes, we do

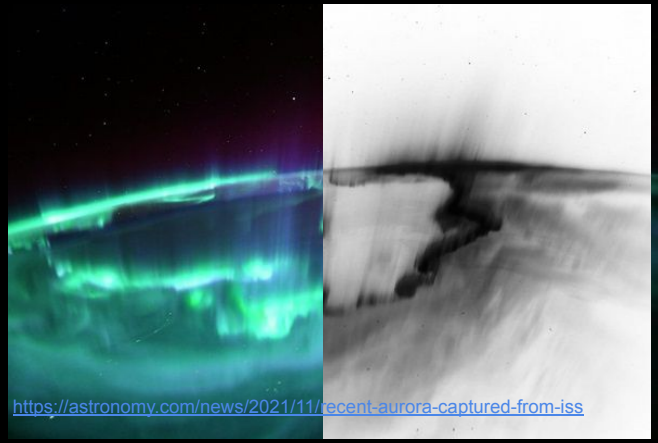
Radio Waves from auroral sources



LaBelle, J., and Anderson, R.R., Ground-level detection of Auroral Kilometric Radiation

How to study these phenomena?

- Where do these emissions come from, and what are their characteristics? Could they be explained by existing theories?
- How can we more directly observe the radio aurora?

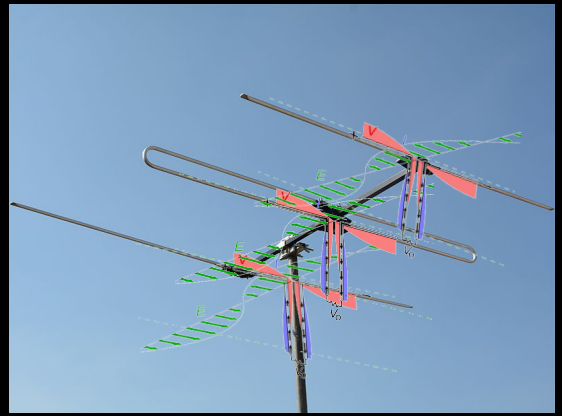


Optical aurora

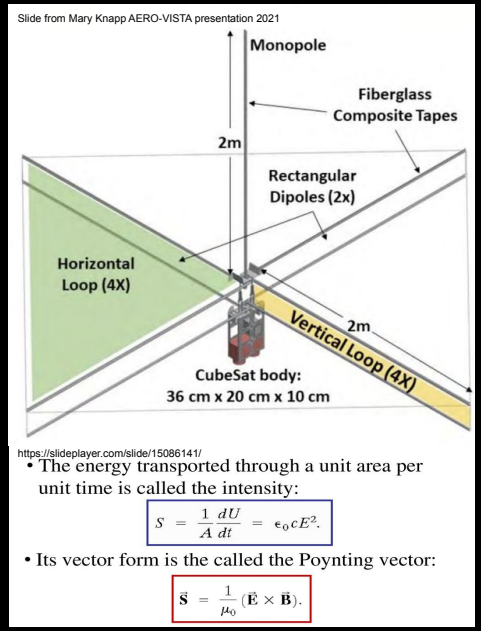
Radio aurora??

The Electromagnetic Vector Sensor

- Satellites above the ionosphere can get us clear access
- Single-element sensing of all six EM vector components; further processing yields polarization and incidence angle
- Allows testing theories of how these emissions are created



Dipole/loop antenna = 1 dimension of data 😔



Vector sensor antenna = 6 dimensions of data 🤩

Auroral
Emissions
Radio
Observatory

Vector
Interferometry
Space
Technology using
AERO

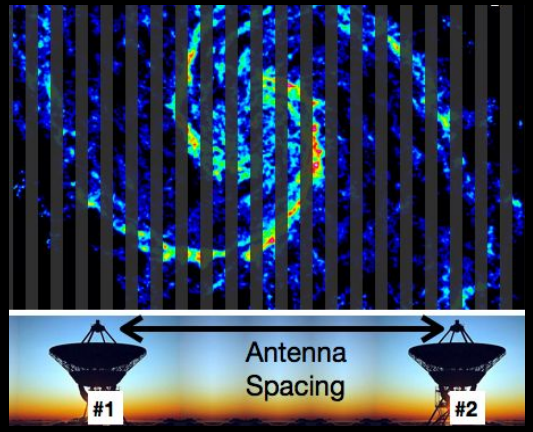
MIT HAYSTACK OBSERVATORY

NASA

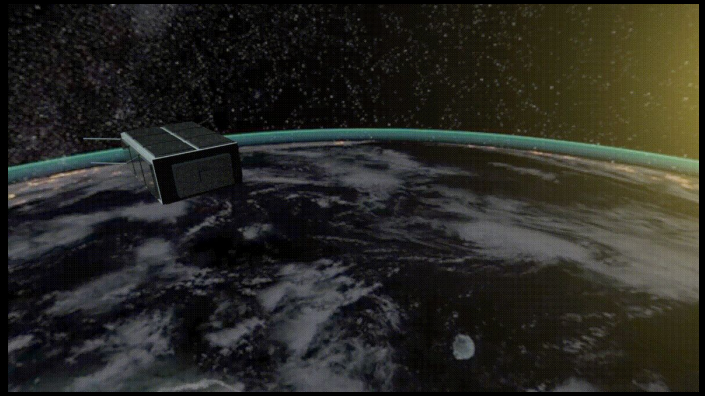
GRCon 22 WASHINGTON, D.C.

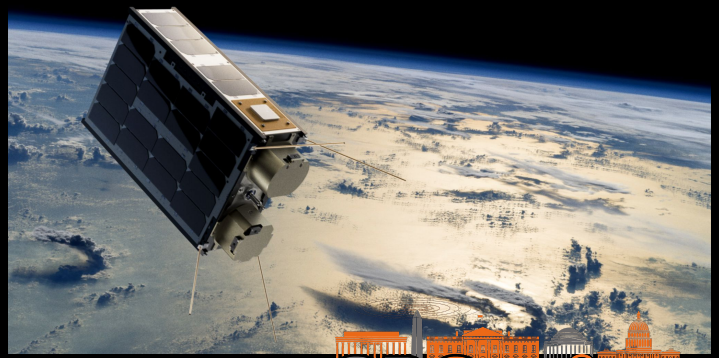
The spacecraft

- Twin 6U CubeSats to perform interferometry in space, significantly increasing angular resolution
- Temporal and spatial detail required means the satellites generate intense amounts of IQ data, which will need to be sent to the ground for further processing
- Very limited bandwidth, so satellites need to be be selective in sending mission data



https://astrobites.org/2013/04/17/the-whirlpool-galaxy-like-youve-never-seen-it-before/





AERO/VISTA Introduction

Ground Operations

Haystack GSI Overview

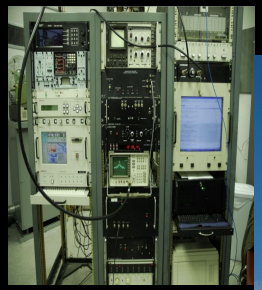
Routing Data with GNURadio

Future Development

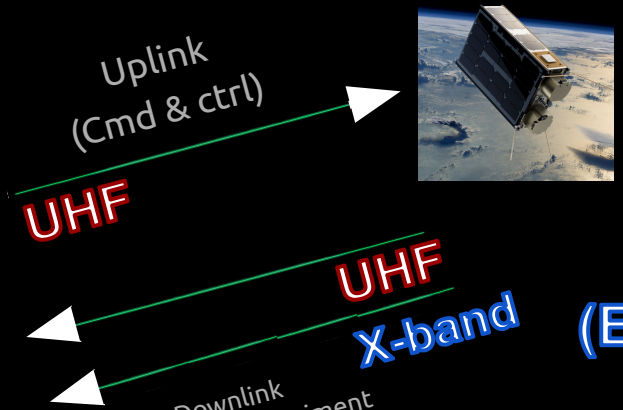
OBJECTIVE #1: We need to talk to the spacecraft.

- Uplink (data to sat): Pings, commands, data acknowledgements; lower-volume command data only
- Downlink (data from sat): Health/status telemetry, experiment data, command acknowledgements; low-volume telemetry and high-volume experiment data

SOLUTION: Use a ground station:



*not the actual ground station



UHF (Telemetry)

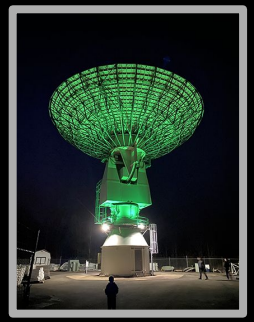
~440 MHz,
Order of 10kbit/s

X-band (Experiment)

~10 GHz,
~20Mbit/s

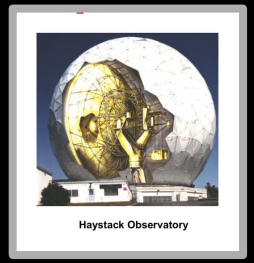
MIT HAYSTACK OBSERVATORY

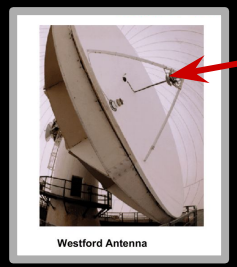
The Ground Station Hardware ✅

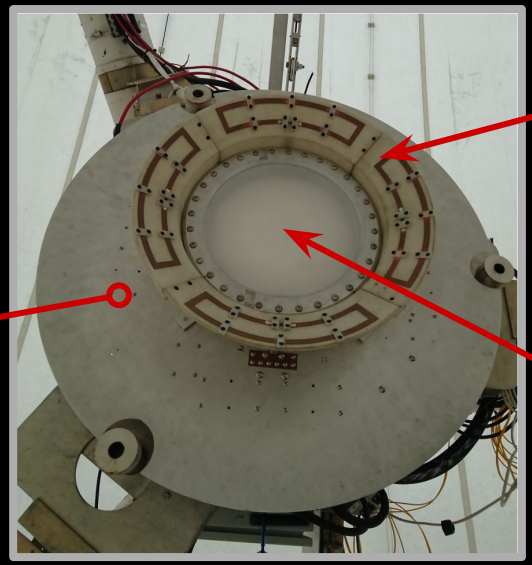


← 21m reflector @ Morehead State University

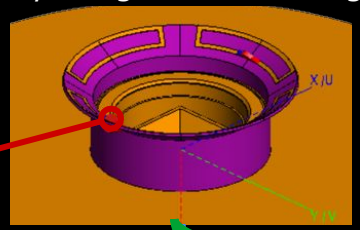
18.6m reflector @ MIT Haystack ↓



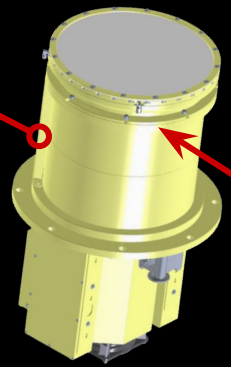




UHF Satcom and Radar Phased Array Feed (TX/RX digital beamforming)



Ettus N310's



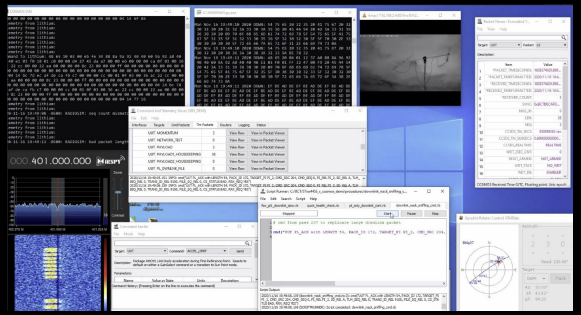
Quad-Ridge Feed Horn w/ Cryo LNA RX only (2 to 16 GHz)

MIT HAYSTACK OBSERVATORY

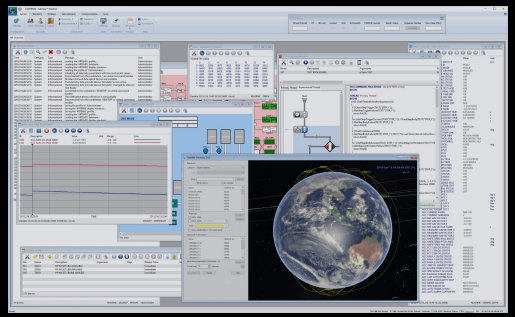
OBJECTIVE #2: We should use a scheduler to smooth the communications process.

- LEO satellite passes only last from 5 to 15 minutes - very limited time to exchange data & send commands
- It's inefficient and unnecessary to have a human manually control low-level communication processes.

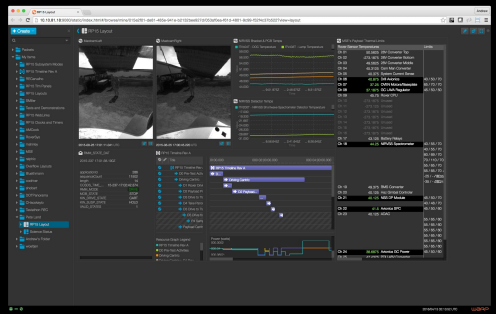
SOLUTION: ??????



One-off mission software solution
(photo credit: MIT DeMi)



Proprietary Tracking Software
$$$$$$$$$$



NASA OpenMCT
Only designed for mission ops

MIT HAYSTACK OBSERVATORY

No easily configurable pre-built solutions

AERO/VISTA Introduction

Ground Operations

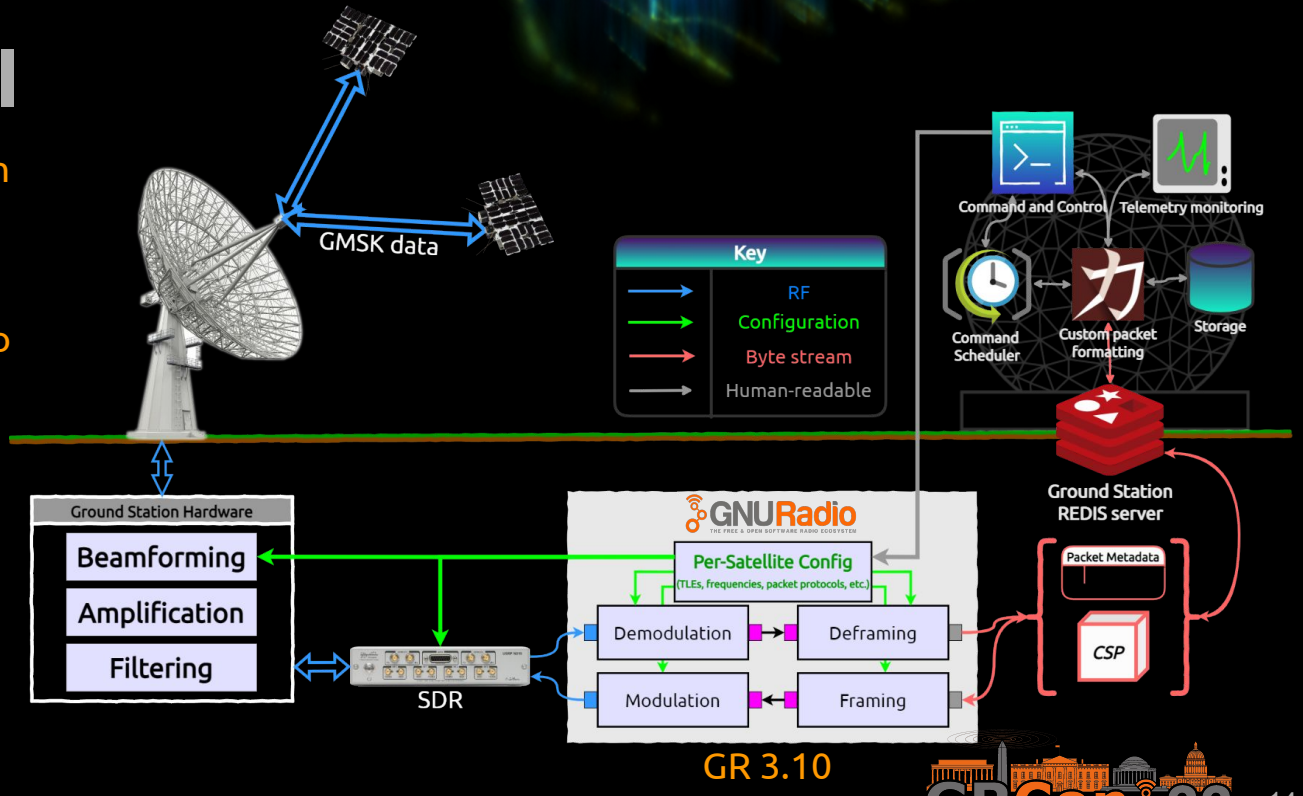
Haystack GSI Overview

Routing Data with GNURadio

Future Development

Haystack GSI

- Will be released as open source software package
- Full stack: from signal to spreadsheet
- Configurable for any satellite or mission
- Keeps critical functions human-readable



AERO/VISTA Introduction

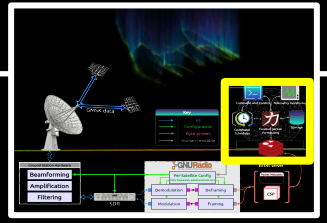
Ground Operations

Haystack GSI Overview

Routing Data with GNURadio

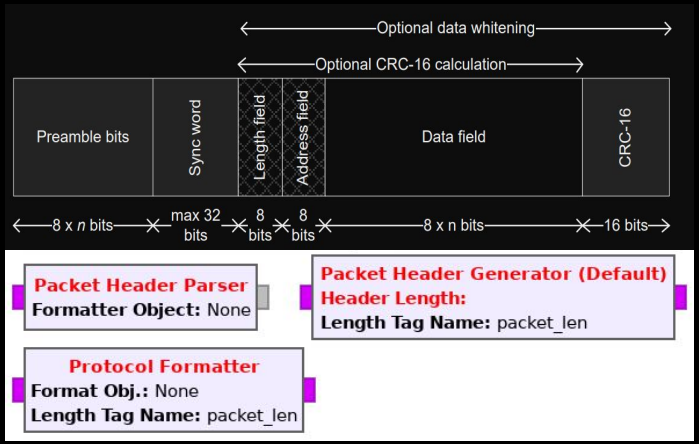
Future Development

MIT HAYSTACK OBSERVATORY



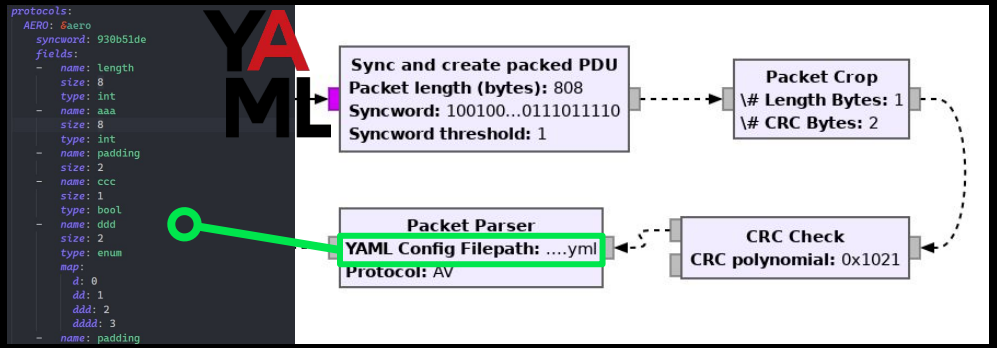
RF Packet processing

How it started



Stream-and-tag based GNURadio packet-parsing blocks

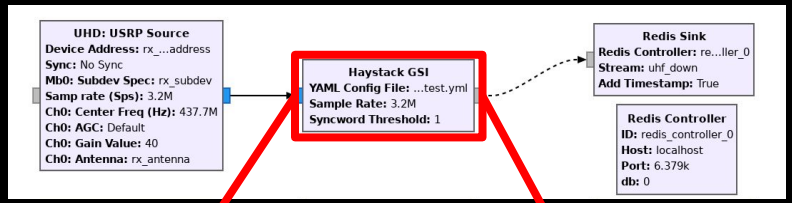
How it's going

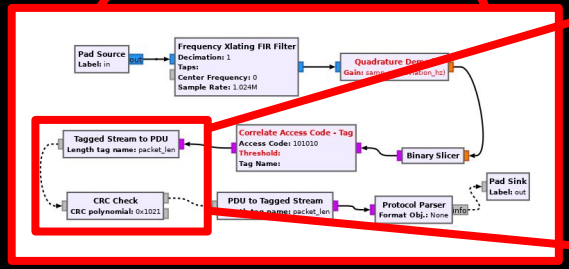


Asynchronous PDU-based decoding; philosophy of "one config to rule them all"

“Hier out” the low-level functions

- Previously unwieldy signal processing chains moved into hot-swappable, predesigned hier blocks
- “Ground station in a box” philosophy: end user should be able to understand minimal signal processing but still be able to up/downlink effectively

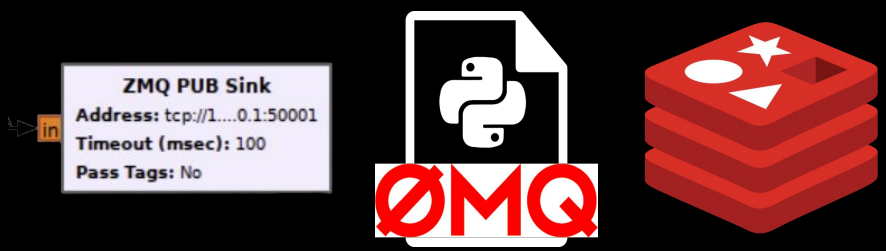





```
radios:
  UHF: &uhf
    source: USRP
    serial: 3094DF7
    up: TX/RX
    down: RX2
    freq_offset: -75000
  digital_rf: &drf
    source: digital_rf
    up: aero-transmissions
    down: aero-test-6
    freq_offset: -75000

channels:
  UHF uplink:
    radio: *uhf
    direction: up
    frequency: 437.7e+6
    modulation: FSK
    deviation: 600
    baudrate: 2400
    protocol: *na
  UHF downlink:
    radio: *uhf
    direction: down
```

Case Study: REDIS import/export

How it started



How it's going

Want to have as few "moving parts" as possible

AERO/VISTA Introduction

Ground Operations

Haystack GSI Overview

Routing Data with GNURadio

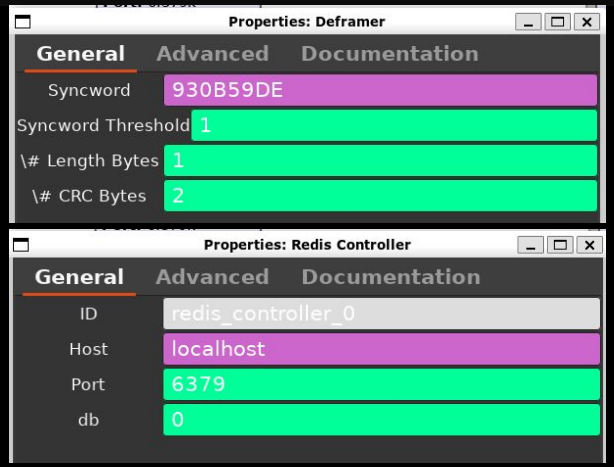
Future Development

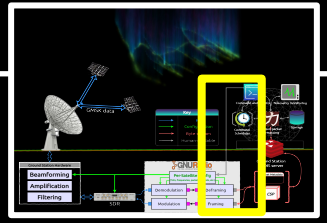
GRC as a Ground Station UI

End goals:

- Program can be run with as little as a few keystrokes
- With a few more clicks, lower level parameters are easily accessible
- Uplink/downlink packets can be routed to different network locations (local redis server, remote host) without dealing with ZMQ or other intermediary transfer protocols: “path of least resistance”
- Hopefully porting everything over to C++ for code efficiency and buffer control

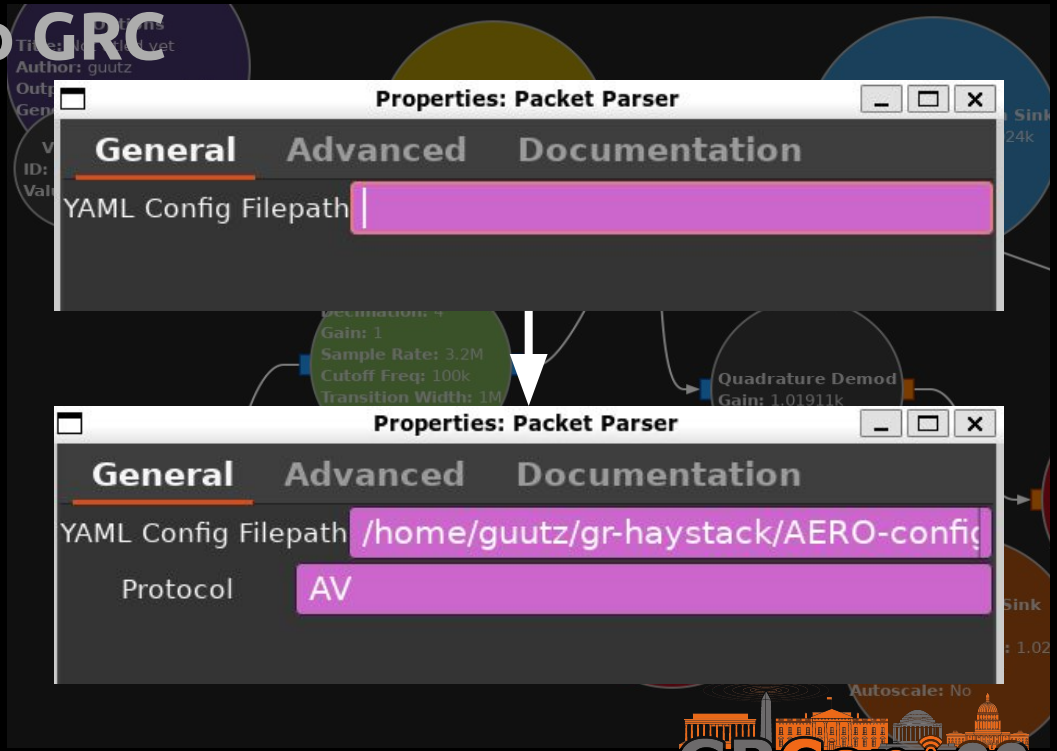
```
~$ gsi.py -c aerovista.yml
```





Extra improvements to GRC

- Dark mode!
- Color- / shape-coded blocks
- Expanded UI and versatility of GRC Blocks
  - Dynamic updating of available options
  - Realtime syncing with a YAML file



Thank you!

Maximilian Riccioli
maximilianriccioli@utexas.edu

Michael Gutierrez
magutier@caltech.edu

REU Grant sponsored by NSF AST-1950348

Special thanks to Dave Stevenson, Rob Phillips, and Freddy Mora at Caltech
Sponsored by Caltech's Tombrello Prize for Undergraduate Physics Research

GRCon primary contacts: Neel Pandeya, Marc Lichtman, Joshua Mormon, Derek Kozel

A-V REU 2022 mentors: Mary Knapp, John Swoboda, Ryan Volz, Tobias Gedenk
A-V Principal Investigators: Phil Erickson, Frank Lind [MIT Haystack]
A-V REU 2022 student collaborators: Allen Chang, Alexis Lupo

Grants: AERO: NASA 80NSSC18K1677; VISTA: NASA 80NSSC19K0617

MIT HAYSTACK OBSERVATORY

GRCon 22 WASHINGTON, D.C.

Resources and Further Reading

AERO: Auroral Emission Radio Observer (2018 Erickson et al.)

AERO & VISTA: Demonstrating HF Radio Interferometry with Vector Sensors (2019 Lind et al.)

Alan Fenn's *Electromagnetics and Antenna Technology*

PySDR - A Digital Signal Processing degree in 2 weeks, guaranteed!™

GitHub - ryanvolz/radioconda: Software radio distribution and installer for conda

GitHub - jopohl/urh: Universal Radio Hacker: Investigate Wireless Protocols Like A Boss

GitHub - MITHaystack/digital_rf: Read, write, and interact with data in the Digital RF and Digital Metadata formats

GitHub - daniestevez/gr-satellites: GNU Radio decoder for Amateur satellites

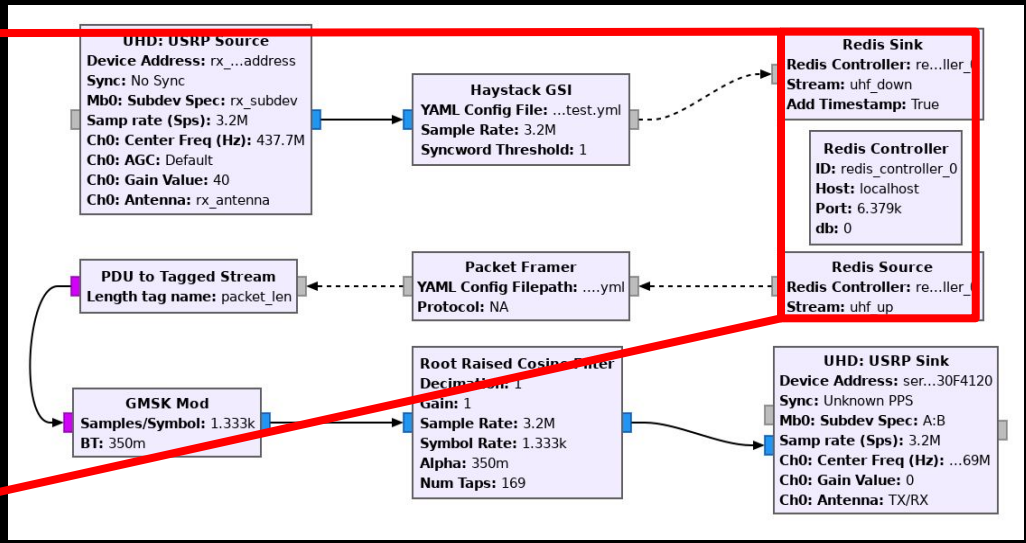
Case Study: REDIS import/export

How it started

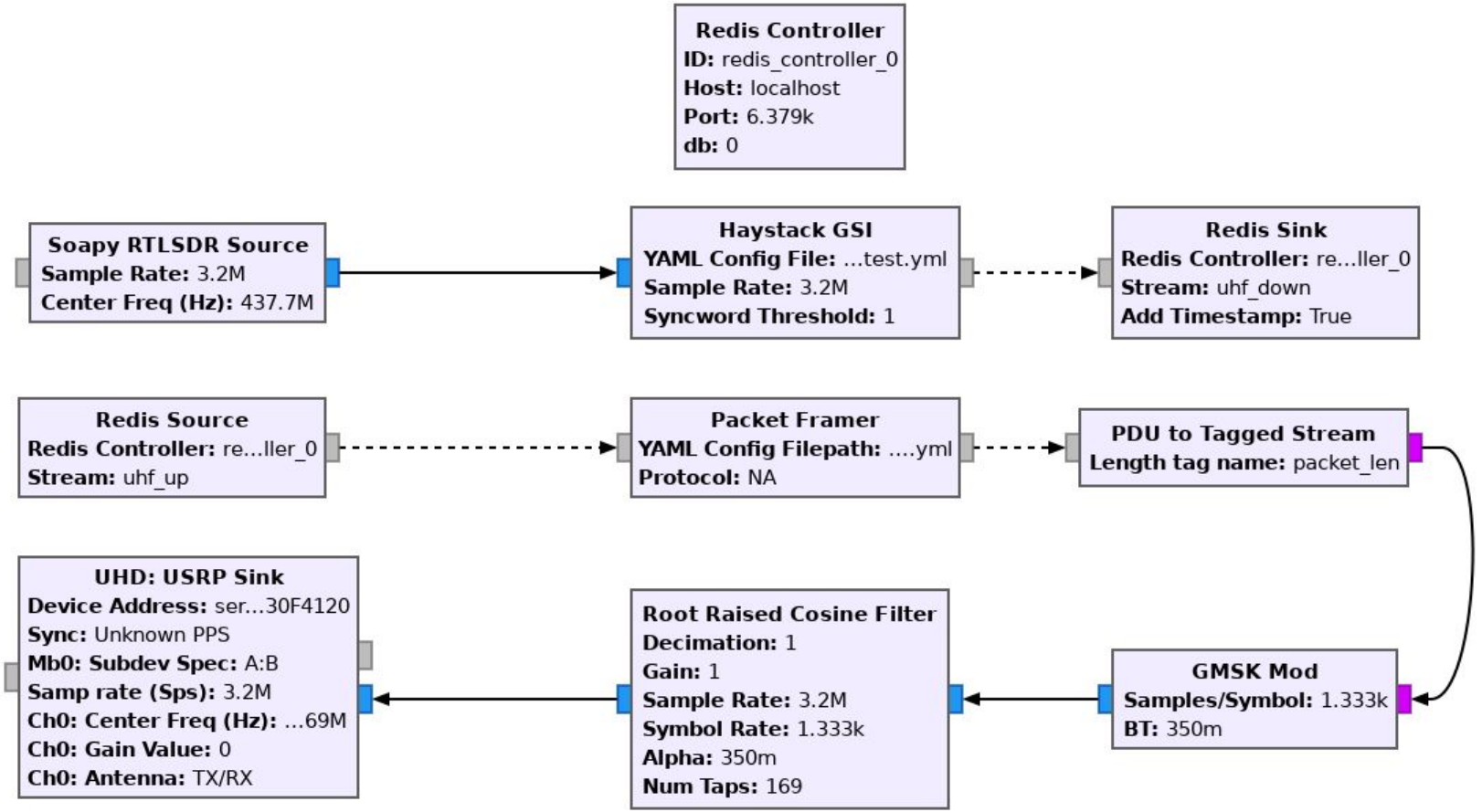
```
class RedisUplink:
    """Handles the REDIS stream dedicated to queueing and collecting packets for transmission over the satellite uplink.
    """
    def __init__(self, config):
        """Initializes the RedisUplink with configuration args in a given dictionary.

        Args:
            config (dict): must include "host", "port", "db", and "pkt_uplink_stream_name"
        """
        self.r = redis.StrictRedis(config["host"], config["port"], config["db"])
        self.uplink_stream_name = config["pkt_uplink_stream_name"]
        def _receiver_target(in_queue: Queue, out_queue: Queue, config_yml_path: str):
            config = GroundStation._get_config(config_yml_path)
            lowgraph = FlowgraphHandler(config["radio"])
            receiver = RadioReceiver(
                config["radio"],
                Deframer(config["radio"])
            )
            downlink = RedisDownlink(config["redis"])
```

Only the basic DSP within GRC:
Python wrapper for flowgraph passes data btwn ZMQ & Ground Station Redis server

How it's going



Hundreds of lines of multiprocessing pycode replaced by a few custom blocks
(and only tens of lines of code to convert btwn redis and PMTs!)


Redis Controller
ID: redis_controller_0
Host: localhost
Port: 6.379k
db: 0
Soapy RTLSDR Source
Sample Rate: 3.2M
Center Freq (Hz): 437.7M
Haystack GSI
YAML Config File: ...test.yml
Sample Rate: 3.2M
Syncword Threshold: 1
Redis Sink
Redis Controller: re...ller_0
Stream: uhf_down
Add Timestamp: True
Redis Source
Redis Controller: re...ller_0
Stream: uhf_up
Packet Framer
YAML Config Filepath: ....yml
Protocol: NA
PDU to Tagged Stream
Length tag name: packet_len
UHD: USRP Sink
Device Address: ser...30F4120
Sync: Unknown PPS
Mb0: Subdev Spec: A:B
Samp rate (Sps): 3.2M
Ch0: Center Freq (Hz): ...69M
Ch0: Gain Value: 0
Ch0: Antenna: TX/RX
Root Raised Cosine Filter
Decimation: 1
Gain: 1
Sample Rate: 3.2M
Symbol Rate: 1.333k
Alpha: 350m
Num Taps: 169
GMSK Mod
Samples/Symbol: 1.333k
BT: 350m