LINUX at 20,000 Meters Over the Sea

Joe VanAndel
National Center for Atmospheric Research

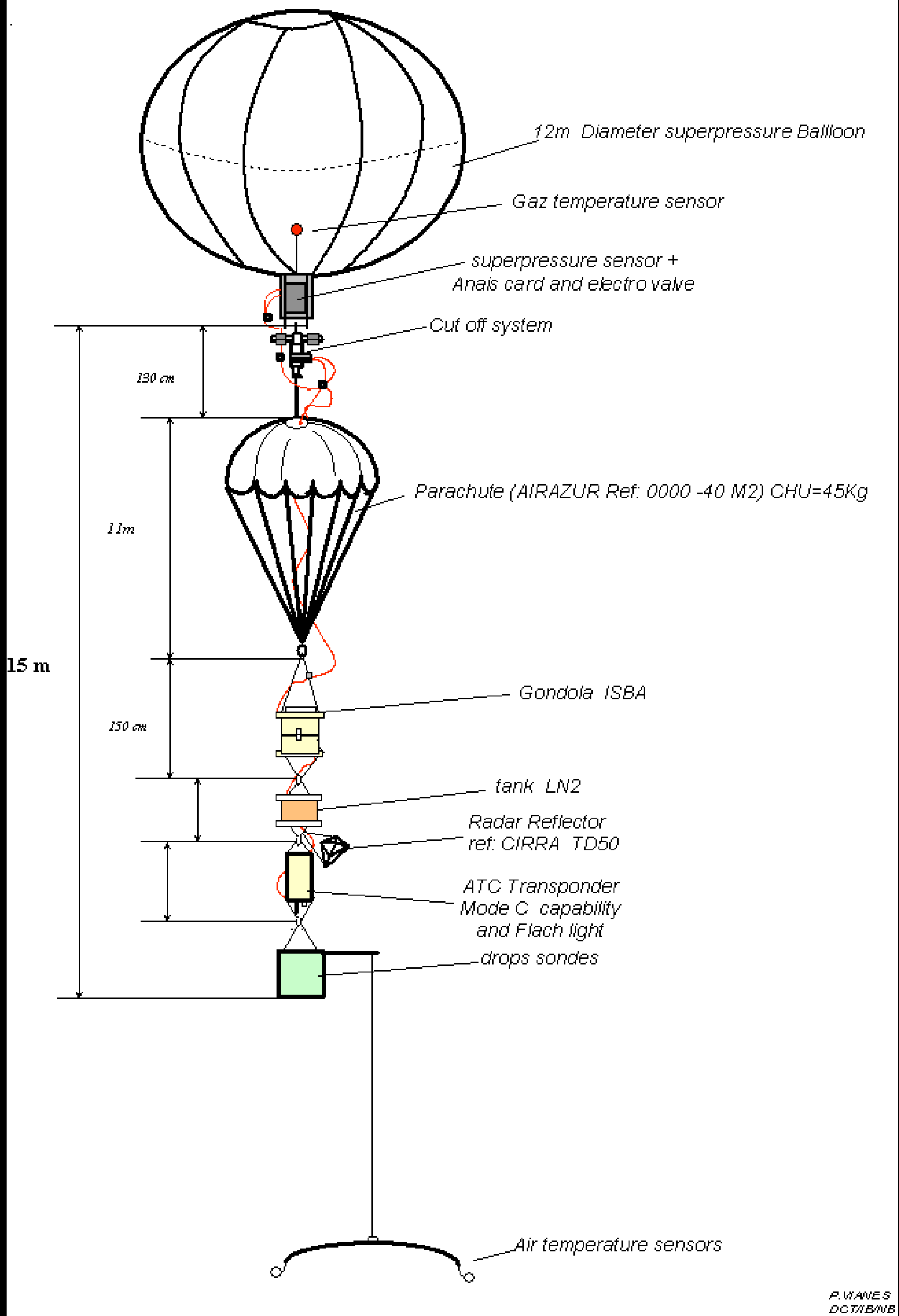
Outline

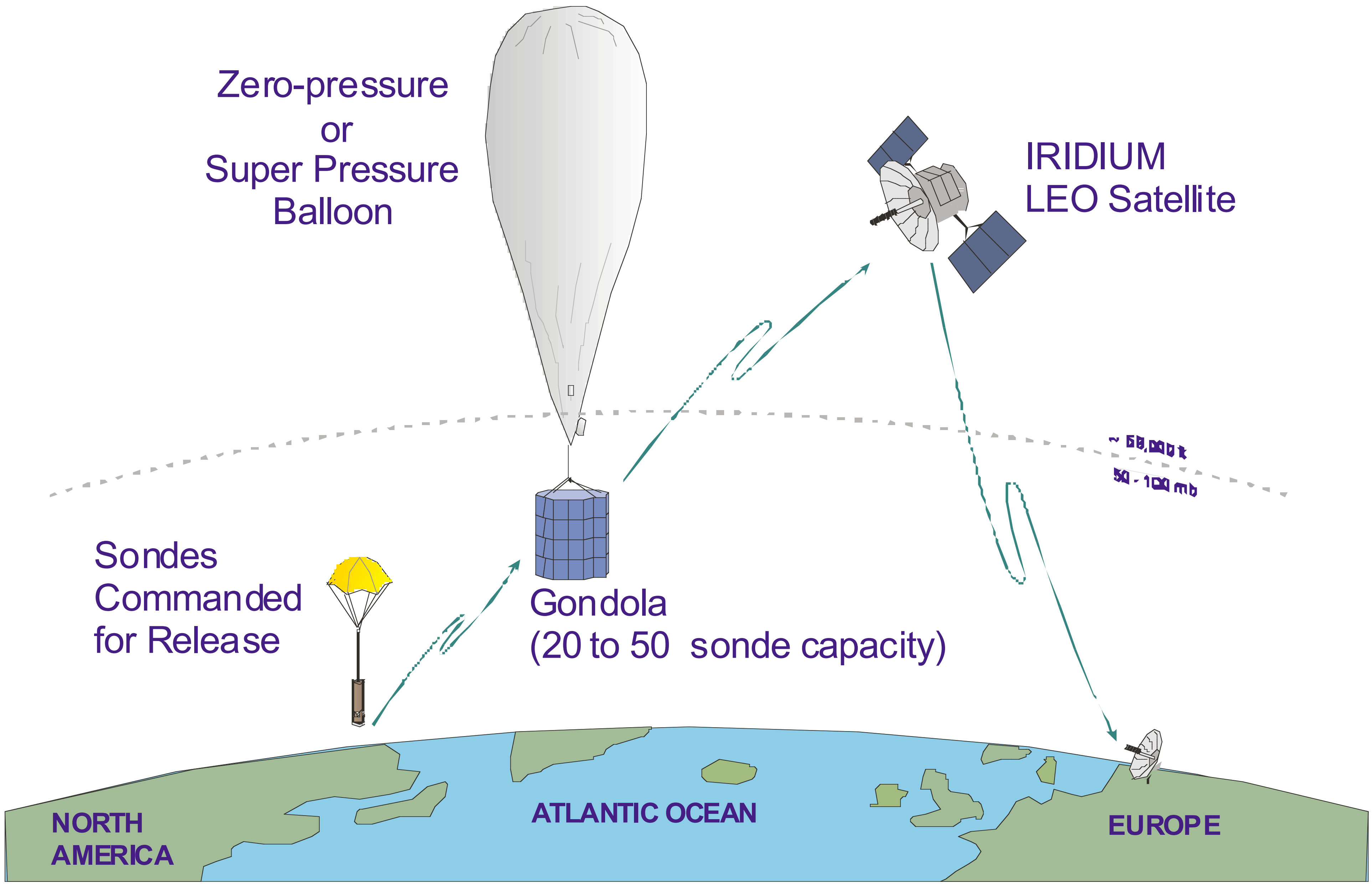
- Introduction - what’s a Driftsonde?
- Who was involved?
- Why did we field Driftsondes?
- How does it work?
- What data did we produce?
- Python Case Study
- What did we learn?

Driftsonde: what is it?

- Stratospheric balloons that fly for 6-12 days.
- Gondola holds ~40 measuring instruments (“sondes”)
- Everything is disposable (wow!)

FLIGHT TRAIN TYPE 2




Zero-pressure
or
Super Pressure
Balloon
IRIDIUM
LEO Satellite
50 - 100 mb
Sondes
Commanded
for Release
Gondola
(20 to 50 sonde capacity)
NORTH
AMERICA
ATLANTIC OCEAN
EUROPE

Who Was Involved?

- National Center for Atmospheric Research (NCAR)
- The French National Space Agency (Centre National d'Etudes Spatiales: CNES)
- French National Center for Scientific Research (Centre National de la Recherche Scientifique: CNRS)

Why did we field Driftsondes?

Driftsonde Goals

- Cost-effective observing system to sample atmospheric data over remote regions with <u>high</u> vertical resolution profiles of:
  - Wind
  - Temperature
  - Humidity

Driftsonde Goals

- Obtaining temperature, humidity, wind speed in “sensitive” areas can improve forecasts for
  - hurricane formation
  - hurricane tracks
  - heavy rain/snow events
- African and Asian weather affects North America’s weather

Driftsonde Goals

- Evaluation/Calibration of Satellite-based techniques:
    - radar measurement of winds
    - water vapor tracking
    - doppler lidar clear air measurements

Why not just use data from satellites?

- Infrared techniques limited by opaque cloud cover
- Microwave techniques have relatively course vertical resolution
- Horizontal winds are a challenge in deep cloud layers

How does it work?

- You need a big balloon


No, not a tiny balloon like this one!

Test Launch in Wyoming

First Launch in Niger, Africa

How does it work?

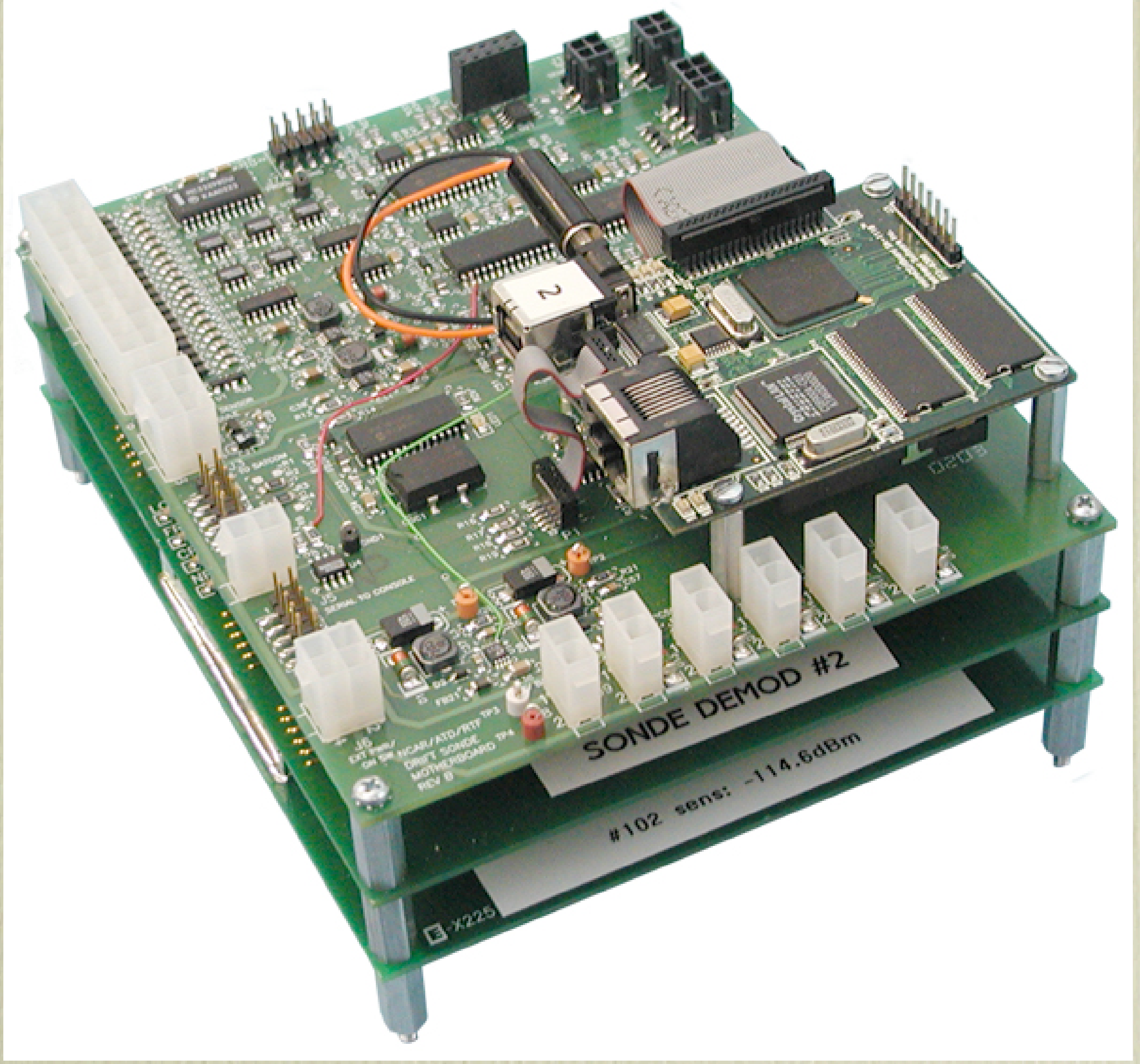
- You need a big balloon
- **You need a gondola (or two!)**

NCAR Gondola

- Onboard computer
- 400 MHz Telemetry Receiver
- 2400 Baud Iridium Satellite modem
- Sondes

Onboard computer

- 400 Mhz XScale computer running Linux
- 32MB flash, 64MB RAM
- 3 serial ports
- Ethernet (for development)
- GPS chip, pressure sensor, voltage and temperature sensors.


SONDE DEMOD #2
#102 sens: -114.6dBm
E-X225

How does it work?

- You need a big balloon
- You need a gondola (or two!)
- **You need to drop sondes**

Sondes: what are they?

- Compact, disposable measurement device:
  - temperature,
  - humidity,
  - wind speed
  - wind direction

Sondes

- Design Constraints:
  - Size - how much weight can we carry?
  - Precision - how accurate is each measurement?
  - Price - how many can we afford?


Aircraft Sondes
MIST
Sonde
FRONTIER

How does it work?

- You need a big balloon
- You need a gondola (or two!)
- You need to drop instruments (sondes)
- **You need to retrieve the data**

Retrieving sonde data

- 400 Mhz, 2400 baud, 1 way radio link from Sonde to Gondola
- Iridium satellite modem (2400 baud) to send data to ground station.

SaVi.sourceforge.net

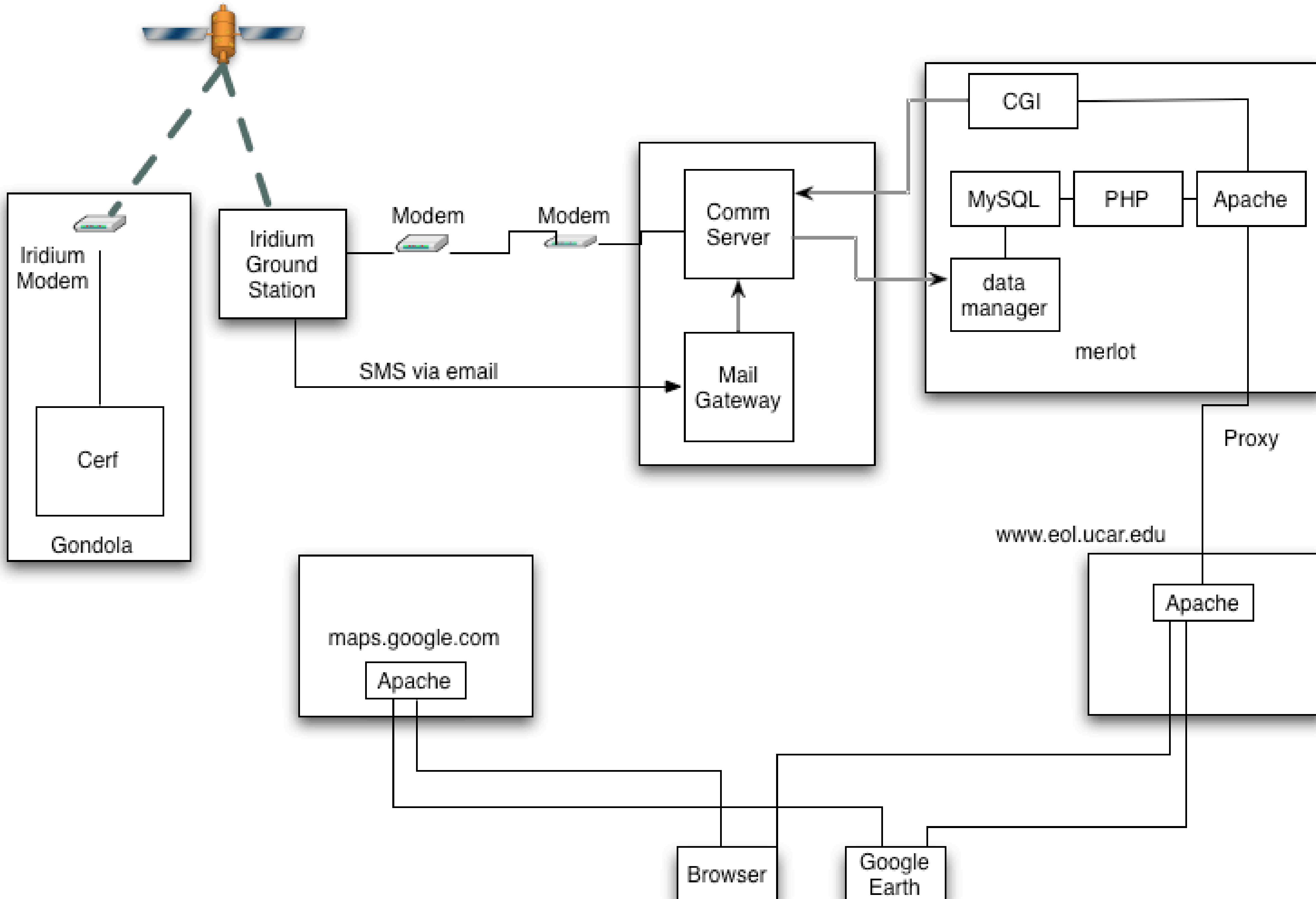
Figure by Lloyd Wood (L.Wood@surrey.ac.uk)

Iridium

- 66 Low Earth Orbiting Satellites
- Civilian ground station in Arizona
- World-wide voice & data: ~$1/minute.
- Data rate is ~2400 baud

How does it work?

- You need a big balloon
- You need a gondola (or two!)
- You need sondes
- You need to retrieve the data
- **You need software to control the system and analyze the data**

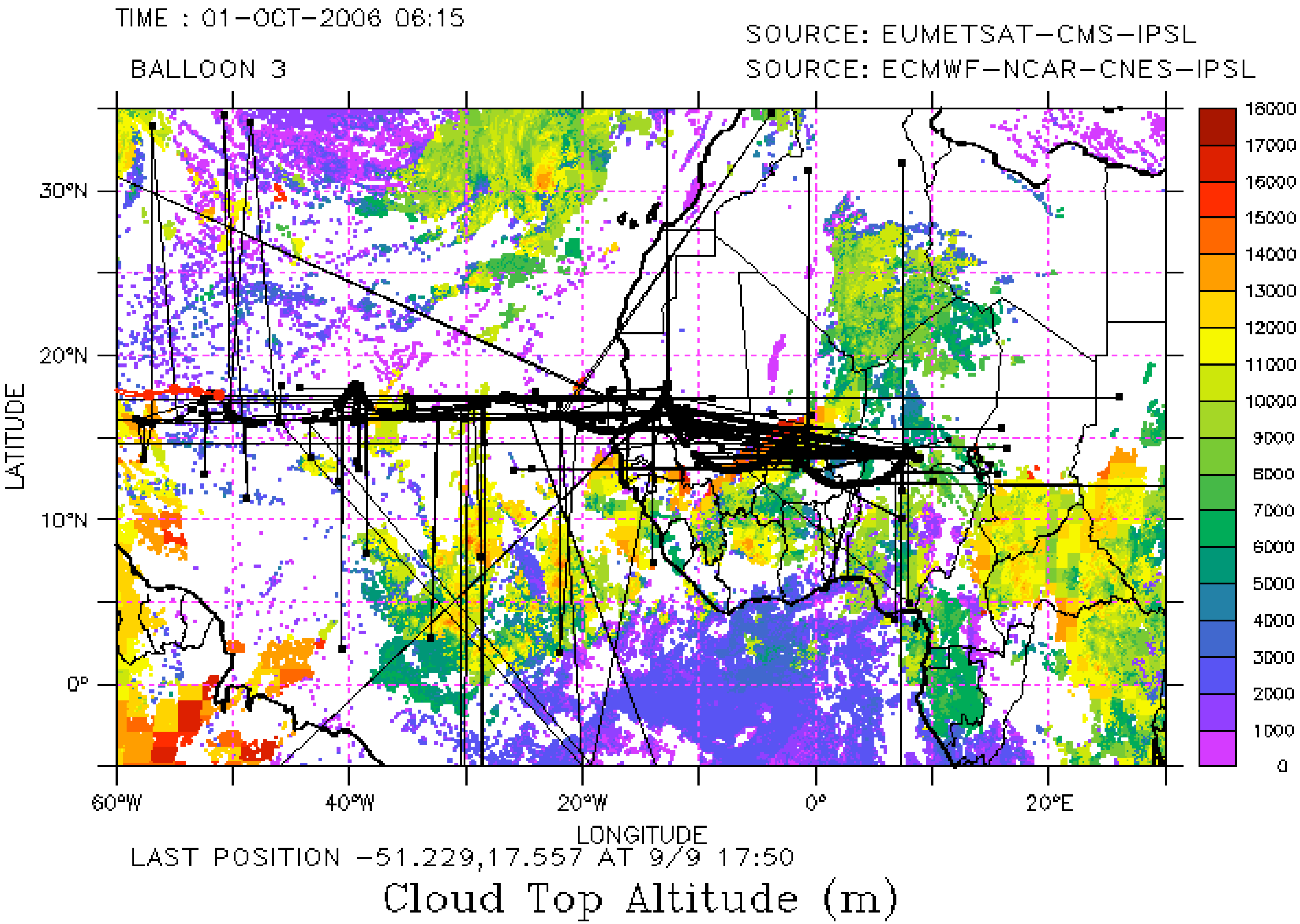


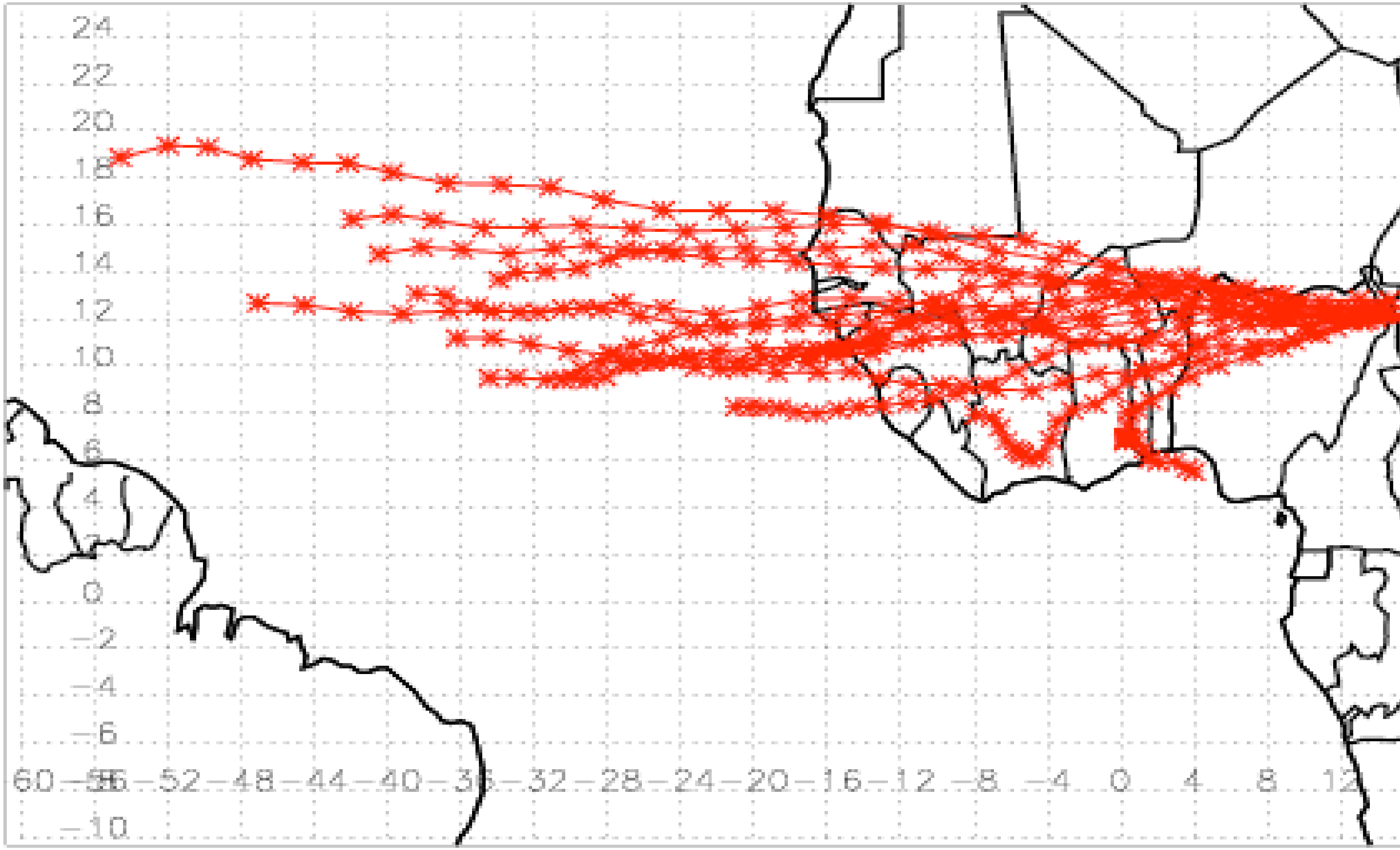
Iridium Modem
Cerf
Gondola
Iridium Ground Station
Modem
Modem
SMS via email
Comm Server
Mail Gateway
CGI
MySQL
PHP
Apache
data manager
merlot
Proxy
www.eol.ucar.edu
Apache
maps.google.com
Apache
Browser
Google Earth

What Data Did We Produce?

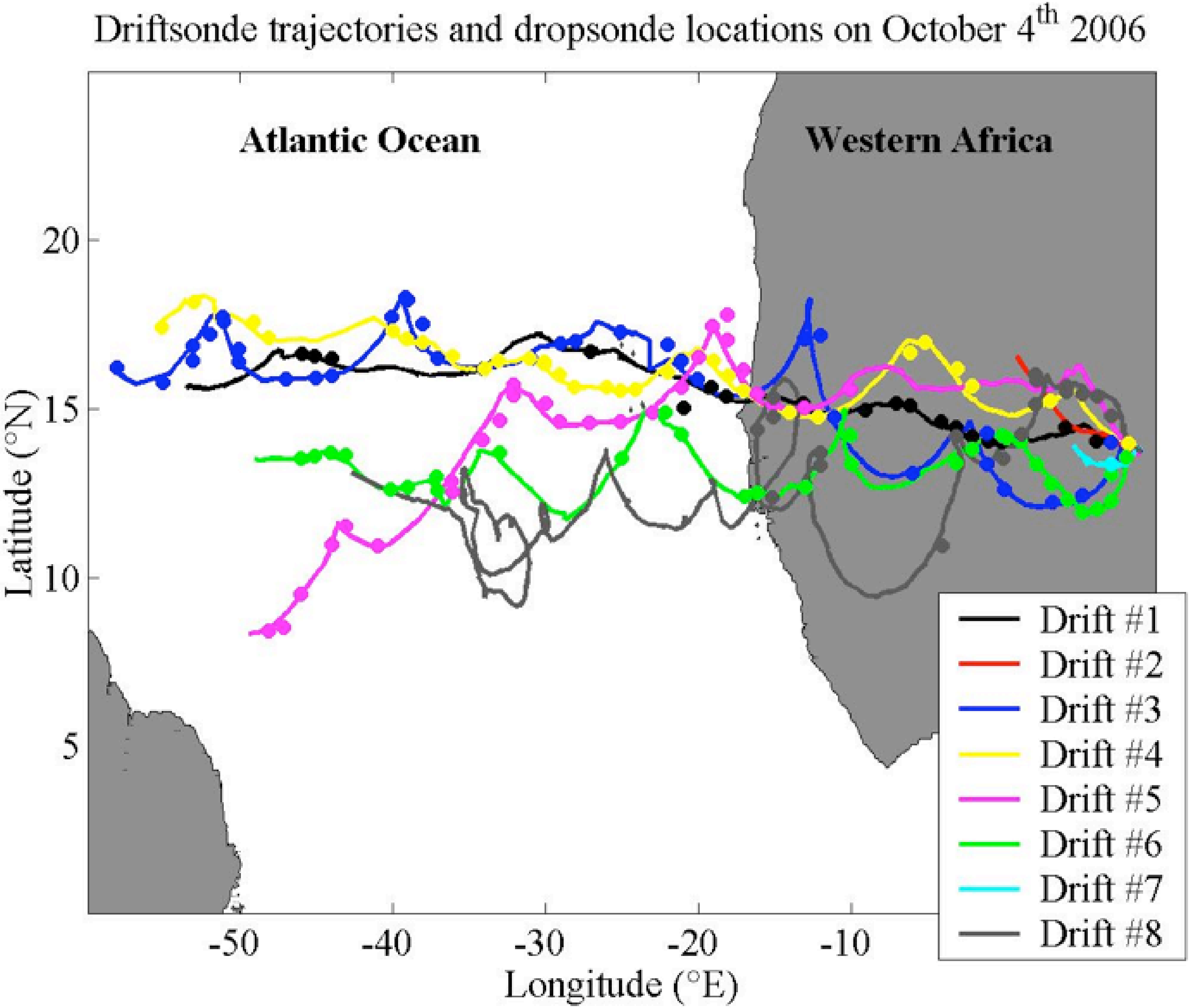
- Geographic Coverage
- Temperature, Humidity, Wind Speed

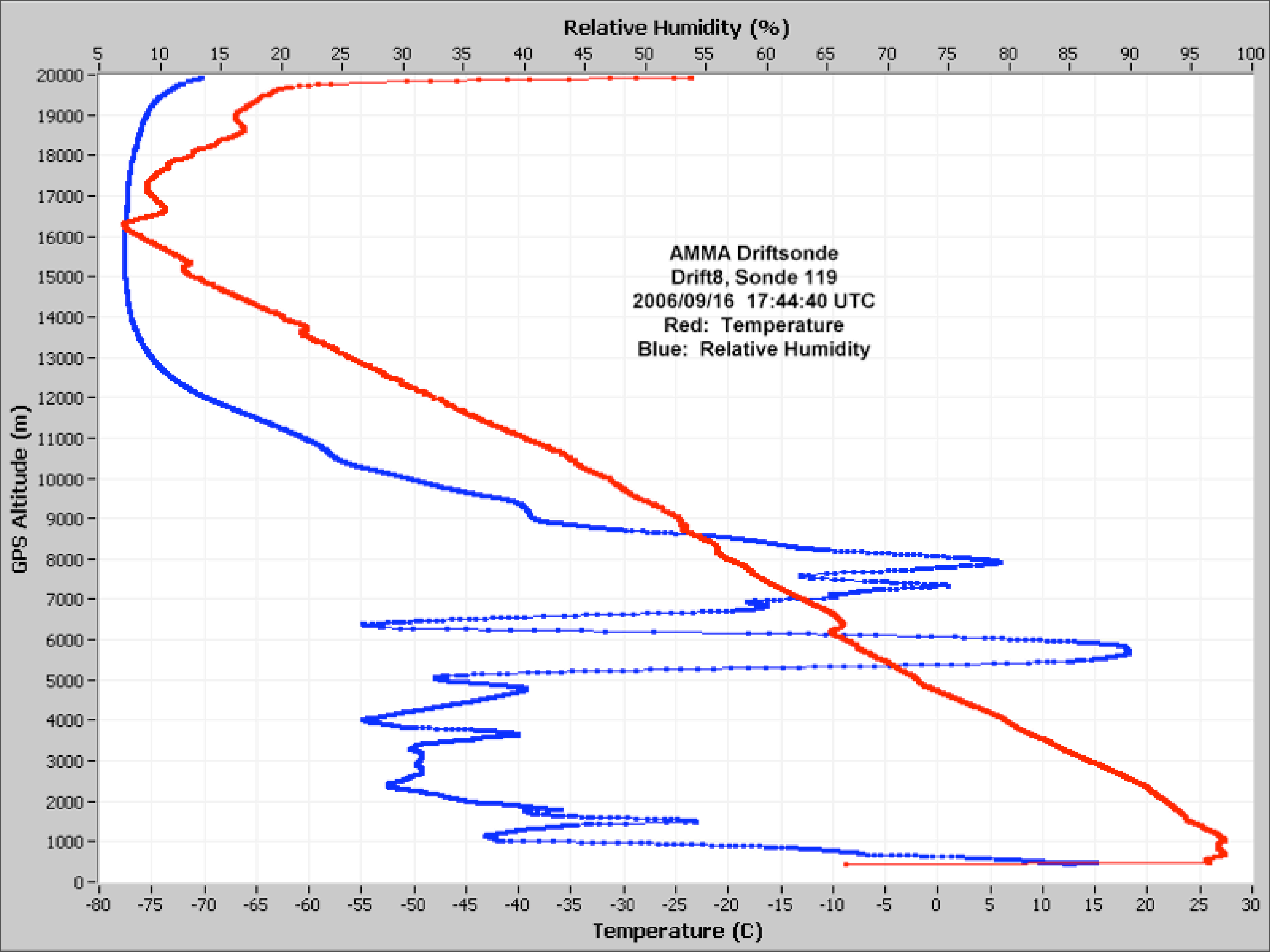
Monitoring Position

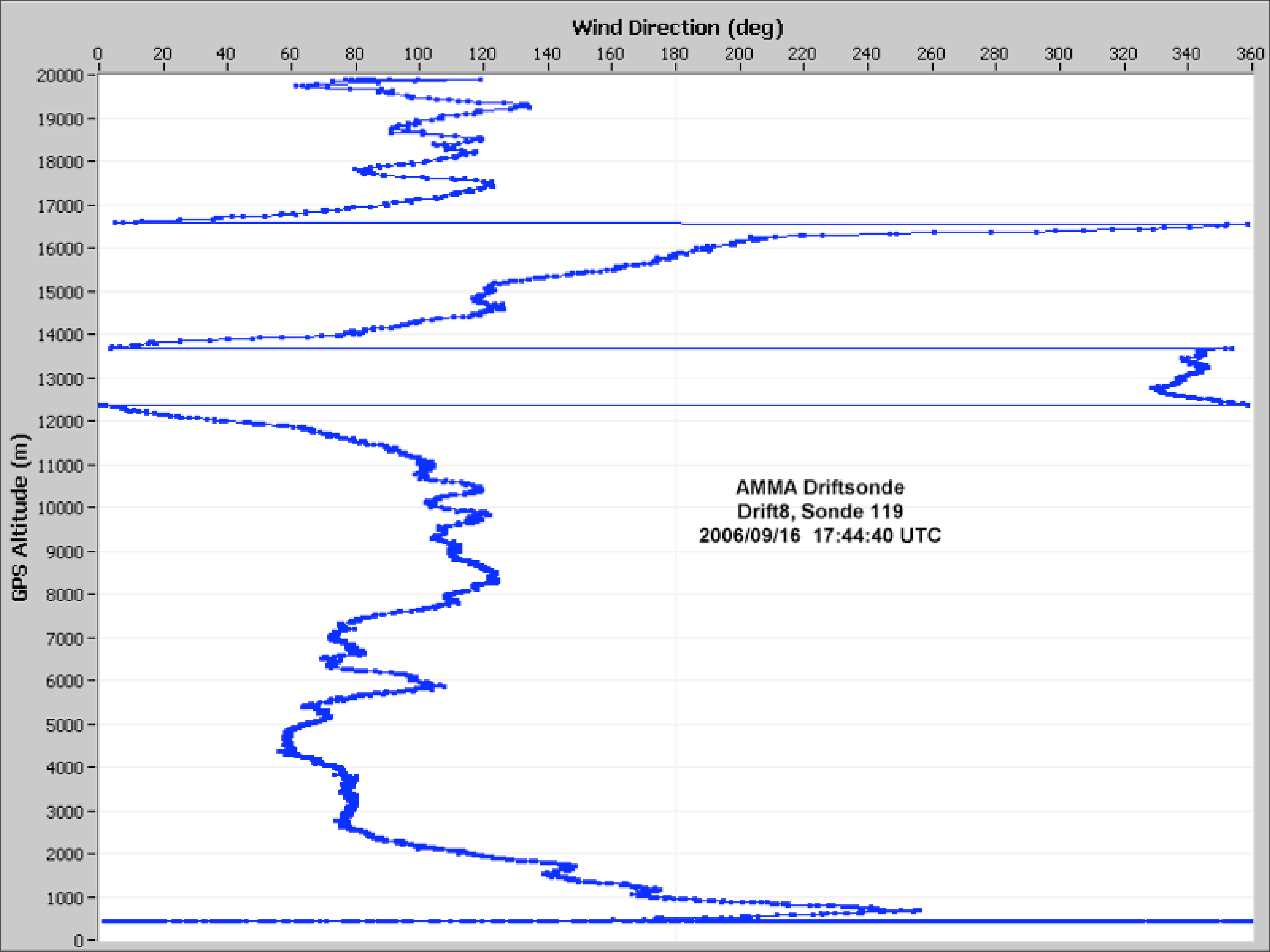




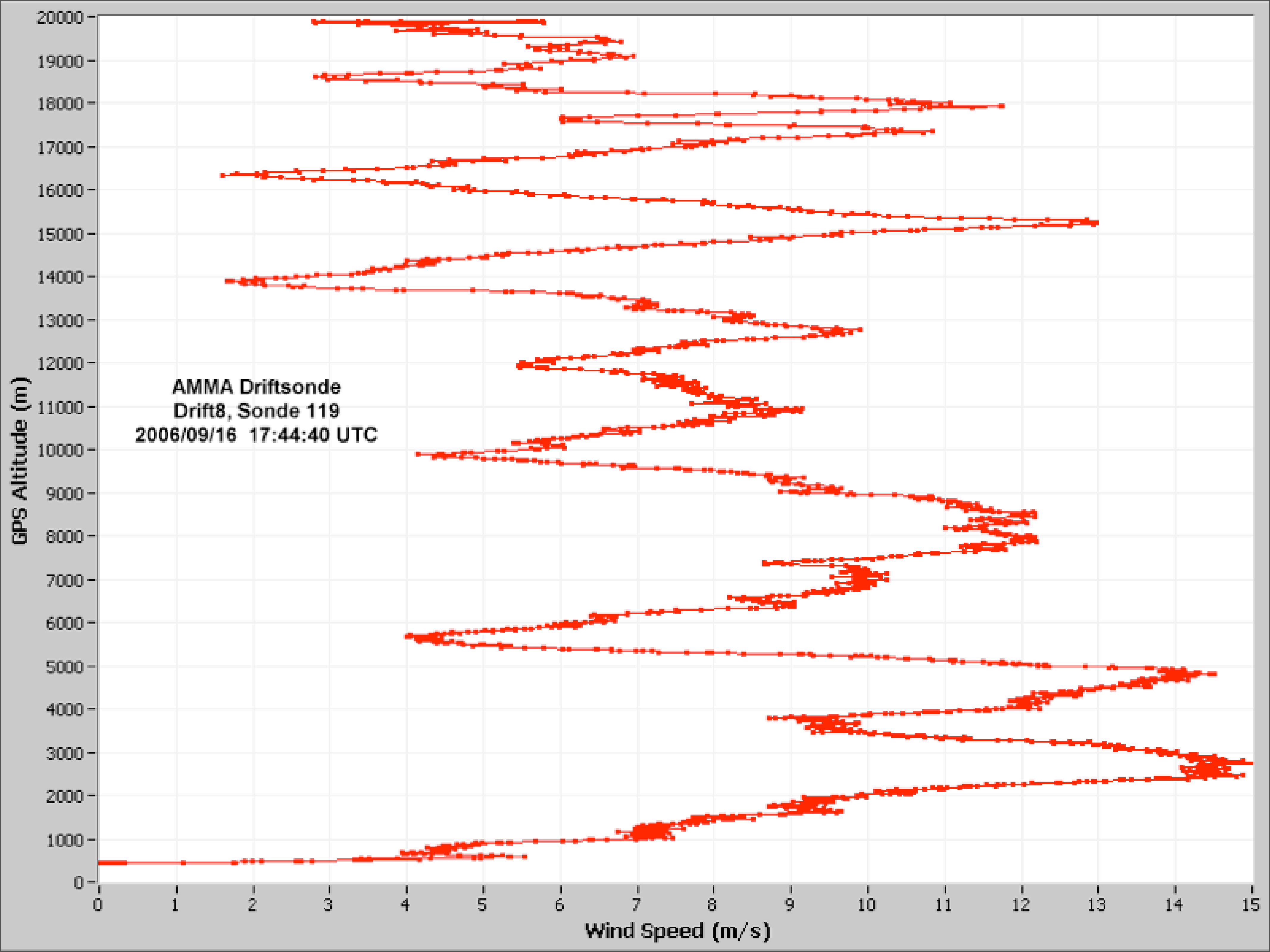
Predicted Trajectories


Driftsonde trajectories and dropsonde locations on October 4th 2006
Atlantic Ocean
Western Africa
20
15
10
5
Latitude (°N)
-50
-40
-30
-20
-10
Longitude (°E)
Drift #1
Drift #2
Drift #3
Drift #4
Drift #5
Drift #6
Drift #7
Drift #8


Relative Humidity (%)
5
10
15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95
100
GPS Altitude (m)
20000
19000
18000
17000
16000
15000
14000
13000
12000
11000
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
AMMA Driftsonde
Drift8, Sonde 119
2006/09/16 17:44:40 UTC
Red: Temperature
Blue: Relative Humidity
-80
-75
-70
-65
-60
-55
-50
-45
-40
-35
-30
-25
-20
-15
-10
-5
0
5
10
15
20
25
30
Temperature (C)



Wind Direction (deg)
0
20
40
60
80
100
120
140
160
180
200
220
240
260
280
300
320
340
360
GPS Altitude (m)
20000
19000
18000
17000
16000
15000
14000
13000
12000
11000
10000
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
AMMA Driftsonde
Drift8, Sonde 119
2006/09/16 17:44:40 UTC


AMMA Driftsonde
Drift8, Sonde 119
2006/09/16 17:44:40 UTC
GPS Altitude (m)
Wind Speed (m/s)
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
11000
12000
13000
14000
15000
16000
17000
18000
19000
20000
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15

Python case study:

- At 20,000 meters : sonde is cold (-70C), needs to be warmed up before it will operate
- Gondola applies power to warm a sonde.
- Discovered that sondes were being warmed up too rapidly - acted flaky!
- We needed to change the heating cycle - in the middle of a flight!

Python case study(2)

- During the implementation phase, I wrote “wrappers” for key C++ classes for Python [SWIG]
- Used the Python wrappers to verify a new heating strategy: 60 seconds on, 15 seconds off.
- Modified the C++ application program to call Python scripts to control heating

Python Case Study(3)

- Uploaded the Python scripts and the modified C++ program to the gondola (~20 minutes)
- Result: could drop some sondes that hadn't worked earlier
- Any additional changes only required editing Python scripts, not recompiling C++ code.

Lessons Learned (1)

- 6 hour hardware reset is invaluable
- software control of peripheral power is quite helpful for error recovery
- simple keyword/value database is great for coordinating tasks
- Python for scripting

Lessons Learned (2)

- Test,Test,Test
- Configuration Management
- Don’t scrimp on essential hardware features - e.g modem control lines:

modem control lines

- Need to know when a modem drops the call - must terminate the shell.
- Had to improvise software solutions to replace the missing modem control lines.

Lessons Learned (3)

- At 2400 baud:
  - short and simple is good
  - command, filename completion is invaluable
  - Patience : but ‘vi’ is still usable

Lessons Learned (4)

- Need additional system status measurements:
  - Sonde temperature
  - Sonde battery voltage (was planned)
  - Gondola power consumption

Lessons Learned (5)

- Always want alternative ways to communicate internationally
  - IRC/Chat
  - Skype
  - Cell phones
  - Iridium voice calls

Lessons Learned (6)

- Wireless access in European hotels
  - Not free : 3 E/hour
  - Reliability, billing can be very frustrating
  - Can be great - login from hotel lobby to a gondola over Africa to check status

Lessons Learned (7)

- For scripting dialup sessions: Kermit is great!
  - Program is mostly free (some restrictions)
  - Bookware: support the project - buy the book - you'll need it!

Future plans

- Web based gondola control
- Gondola tracking via Mapserver
- Use of SMS for gondola status
- Possible experiment from Japan and Hawaii: Fall, Winter 2008
- Possible flights from Antarctica.

Acknowledgements

- NCAR: Keith Romberg, Charlie Martin, Terry Hock, Jack Fox, Hal Cole, Dave Parsons, Dean Lauritsen, Chip Owens, Nicholas Potts, Ken Norris, Steve Deyo
- CNES: Philippe Cocquerez, Stephanie Venel
- CNRS: Philippe Dobrinski


Questions?