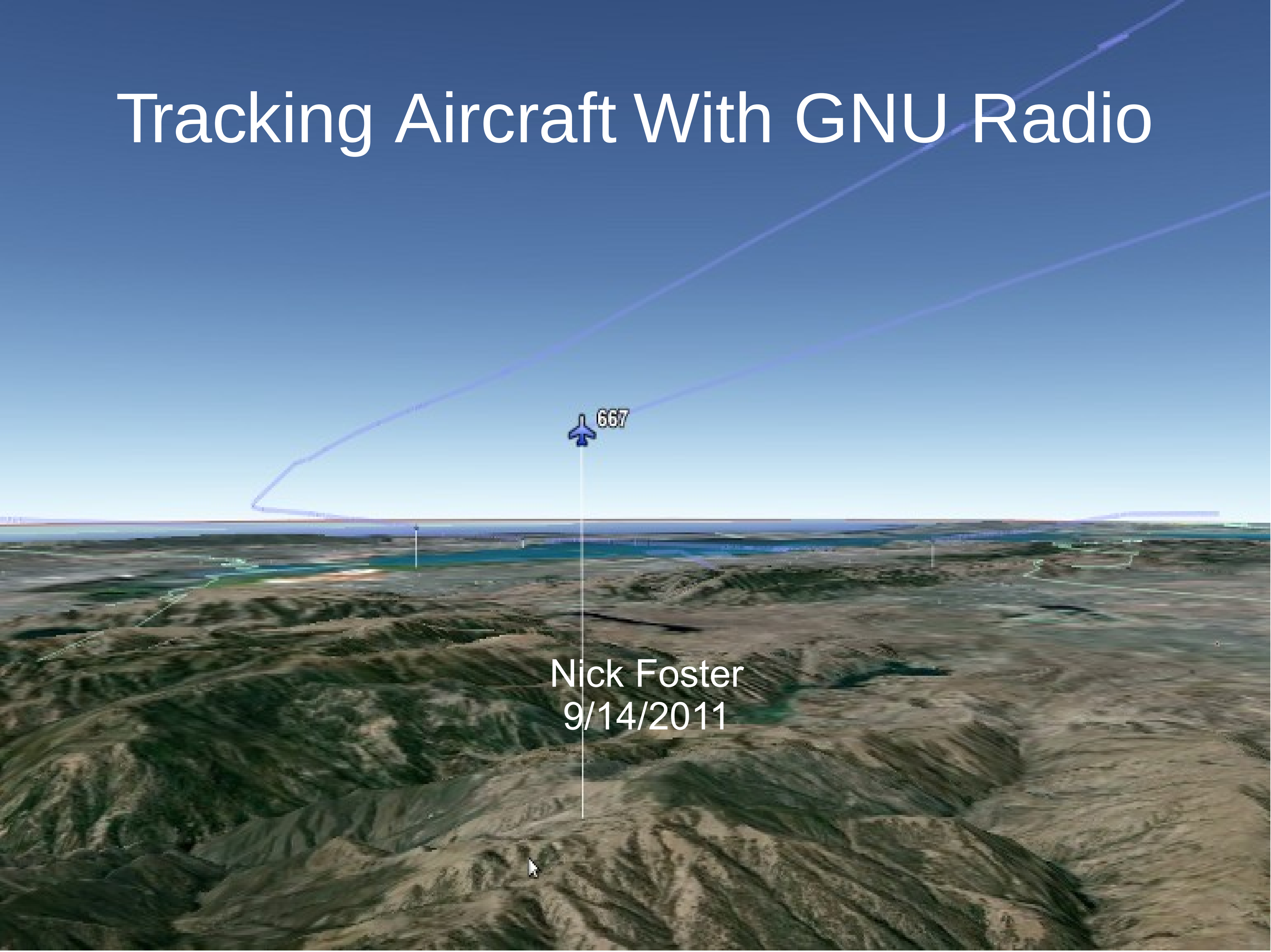
Tracking Aircraft With GNU Radio

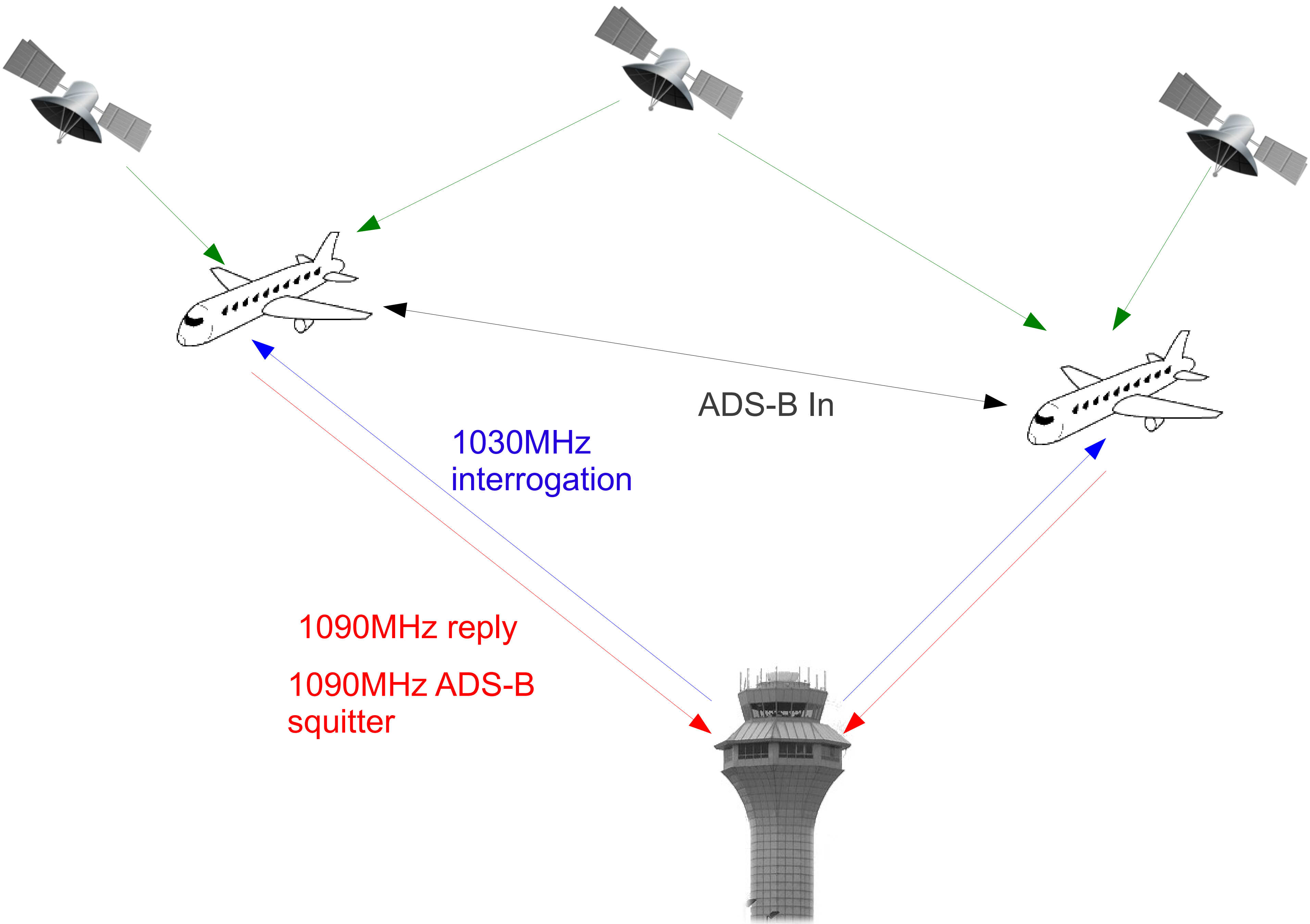


Nick Foster
9/14/2011

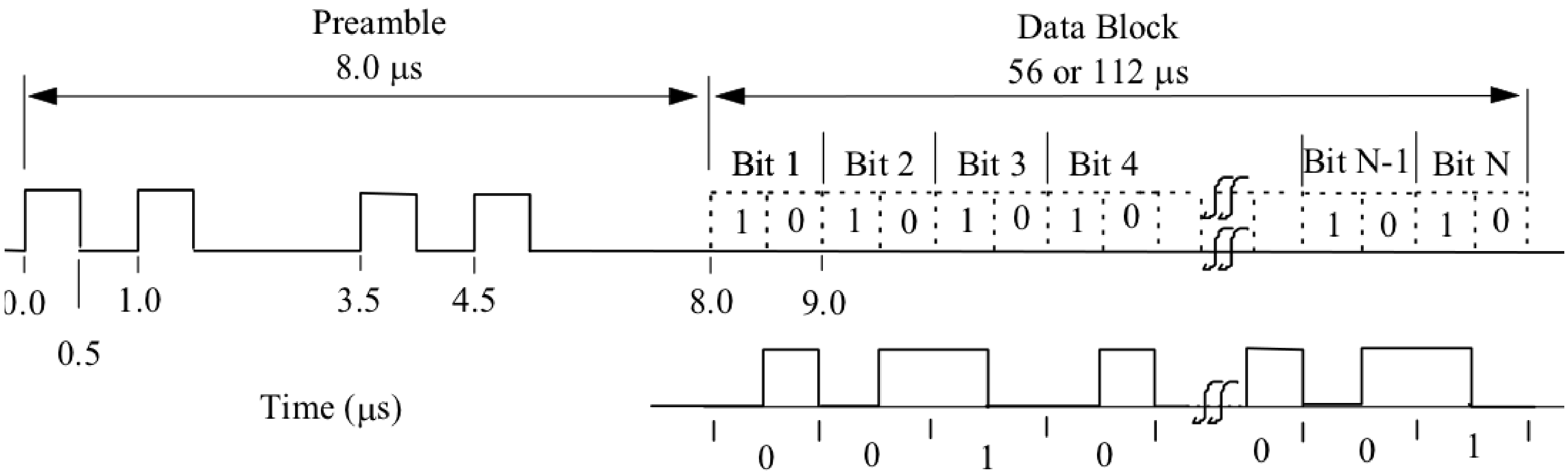
What Is ADS-B?

- **A**utomatic **D**ependent **S**urveillance-**B**roadcast
- ATC of the future (FAA NextGen project)
- Required on most AC in US by 2020, already required in Europe
- Augments primary surveillance radar (ADS-B Out)
- Gives pilots their own radar picture (ADS-B In)
- Transport layer, not physical (OSI level 4)
- Several PHY layer implementations
  - UAT
  - **Mode S Extended Squitter**
  - VDL modes 2/4

ADS-B enhanced ATC system



The Mode S Waveform



- Pulse position modulation at 1Mbit
- Simple modulation scheme designed to be TX/RX by cheap hardware
- PHY layer compatible with older transponders (Mode A → Mode C → Mode S)
- Uniquely identifies **every** aircraft by airframe ID
- No media access control at all – transponders just step on each other (“FRUIT”)
- ADS-B uses “extended squitter”, 112 bit packets

Receiver overview

- Filtering and AM demod
- Preamble detector
  - Threshold detector followed by validation
  - Tags first sample of preamble packet w/timestamp
  - Only forwards viable samples downstream
    - This corrupts your timestamp sample reference!
- Data slicer
  - Searches for preamble tags
  - Slices bits and produces messages to Python main app
- Packet parsing and output modules harvest message queue
  - Google Earth interface
  - NoSQL database for other custom apps

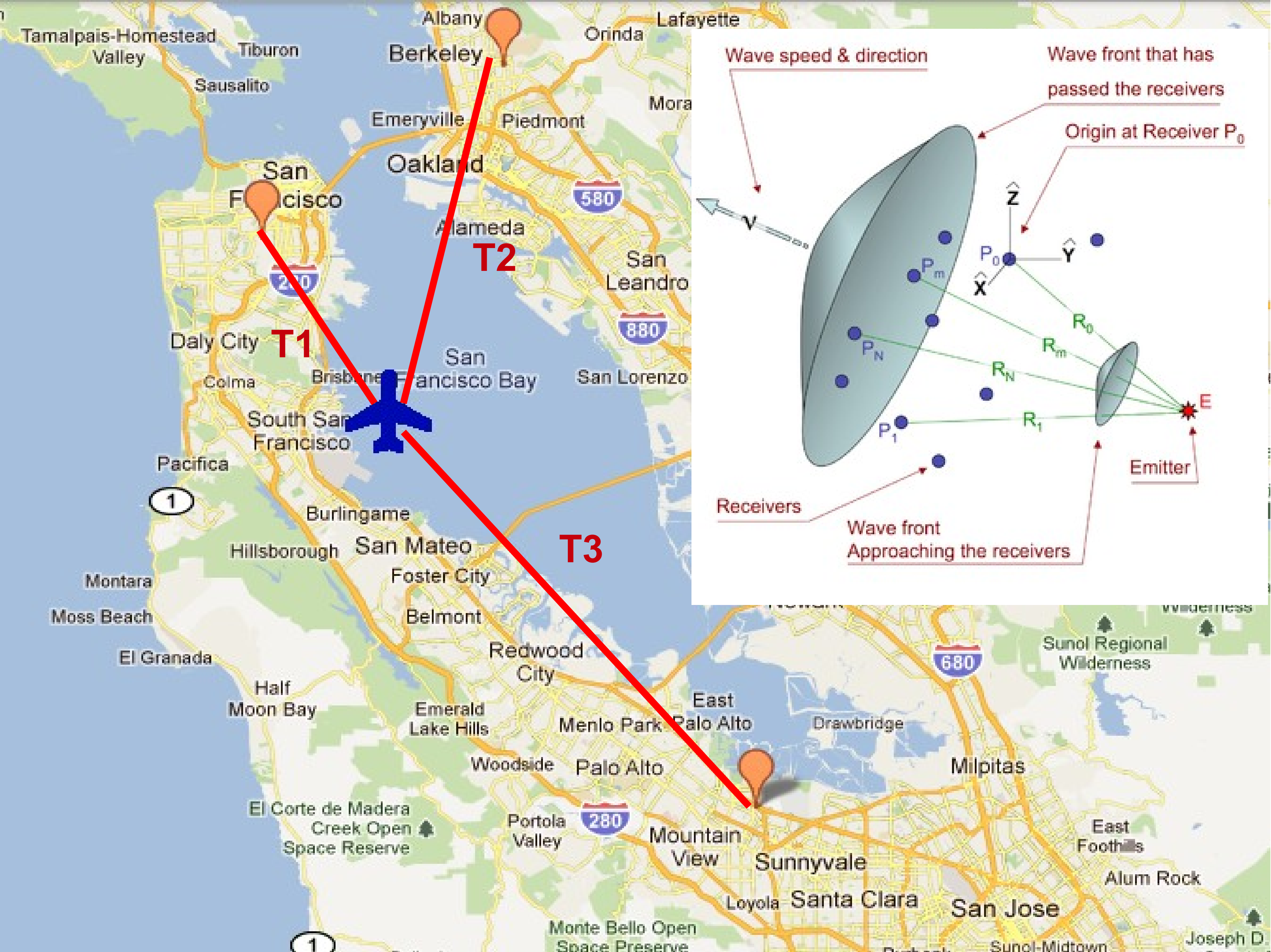
Receiver Performance and Results

- Aircraft successfully tracked from 275 nautical miles away!
- Typical range 100-150nm depending on direction and altitude
- Setup:
  - USRP1
  - DBSRX
  - Mini-Circuits LNA
  - 1090MHz SAW filter
- Interesting aircraft heard – Air Force One, Blue Angels, C-130, AWACS

```
(-24 0 0.000000) Type 17 subtype 04 (ident) from adfdf8 with data AF1
(-26 0 0.000000) Type 4 (short surveillance altitude reply) from adfdf8 at 1125ft
(-25 0 0.000000) Type 0 (short A-A surveillance) from adfdf8 at 1125ft (speed
300-600kt)
(-24 0 0.000000) Type 11 (all call reply) from adfdf8 in reply to interrogator 0
```

Multilateration

- ADS-B is great and all, but only 30% compliance in the US
  - What good is a collision avoidance system which ignores 70% of the traffic?
  - What about interesting, nonparticipating aircraft (AF1, Blue Angels, etc.)?
- Mode S provides a unique airframe ID to differentiate replies
- Hyperbolic positioning based on TDOA – just like GPS



T1
T2
T3
Wave speed & direction
Wave front that has passed the receivers
Origin at Receiver P_0
V
Z
Y
X
P_0
P_m
P_N
P_1
R_0
R_m
R_N
R_1
E
Emitter
Receivers
Wave front Approaching the receivers
Tamalpais-Homestead Valley
Tiburon
Sausalito
Albany
Berkeley
Lafayette
Orinda
Emeryville
Piedmont
Oakland
San Francisco
Alameda
San Leandro
Daly City
Colma
Brisbane
San Francisco Bay
San Lorenzo
South San Francisco
Pacifica
Burlingame
Hillsborough
San Mateo
Foster City
Montara
Moss Beach
Belmont
El Granada
Redwood City
Half Moon Bay
Emerald Lake Hills
Menlo Park
East Palo Alto
Drawbridge
Woodside
Palo Alto
Milpitas
El Corte de Madera Creek Open Space Reserve
Portola Valley
Mountain View
Sunnyvale
Loyola
Santa Clara
San Jose
Sunol Regional Wilderness
East Foothills
Alum Rock
Monte Bello Open Space Preserve
580
880
280
680
1

Maybe it's not like GPS after all

- We get a free bit of knowledge: altitude
  - Lets us multilaterate with three, rather than four stations (albeit with poor geometry)
  - Extra “receiver” located at Earth's center with simulated TDOA to known altitude
- Non-synchronized receivers
  - We can synchronize using ADS-B-equipped aircraft!
    - But we have to cope with clock offset and clock drift
    - Remember, 1 microsecond is 300 meters

The inevitable math

$$\begin{pmatrix} \Delta P^1 \\ \Delta P^2 \\ \Delta P^3 \\ \vdots \\ \Delta P^m \end{pmatrix} = \begin{pmatrix} \frac{\partial P^1}{\partial x} & \frac{\partial P^1}{\partial y} & \frac{\partial P^1}{\partial z} & \frac{\partial P^1}{\partial \tau} \\ \frac{\partial P^2}{\partial x} & \frac{\partial P^2}{\partial y} & \frac{\partial P^2}{\partial z} & \frac{\partial P^2}{\partial \tau} \\ \frac{\partial P^3}{\partial x} & \frac{\partial P^3}{\partial y} & \frac{\partial P^3}{\partial z} & \frac{\partial P^3}{\partial \tau} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial P^m}{\partial x} & \frac{\partial P^m}{\partial y} & \frac{\partial P^m}{\partial z} & \frac{\partial P^m}{\partial \tau} \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta \tau \end{pmatrix} + \begin{pmatrix} v^1 \\ v^2 \\ v^3 \\ \vdots \end{pmatrix}$$

$$\mathbf{A} = \begin{pmatrix} \frac{x_0 - x^1}{\rho} & \frac{y_0 - y^1}{\rho} & \frac{z_0 - z^1}{\rho} & c \\ \frac{x_0 - x^2}{\rho} & \frac{y_0 - y^2}{\rho} & \frac{z_0 - z^2}{\rho} & c \\ \frac{x_0 - x^3}{\rho} & \frac{y_0 - y^3}{\rho} & \frac{z_0 - z^3}{\rho} & c \\ \vdots & \vdots & \vdots & \vdots \\ \frac{x_0 - x^m}{\rho} & \frac{y_0 - y^m}{\rho} & \frac{z_0 - z^m}{\rho} & c \end{pmatrix}$$

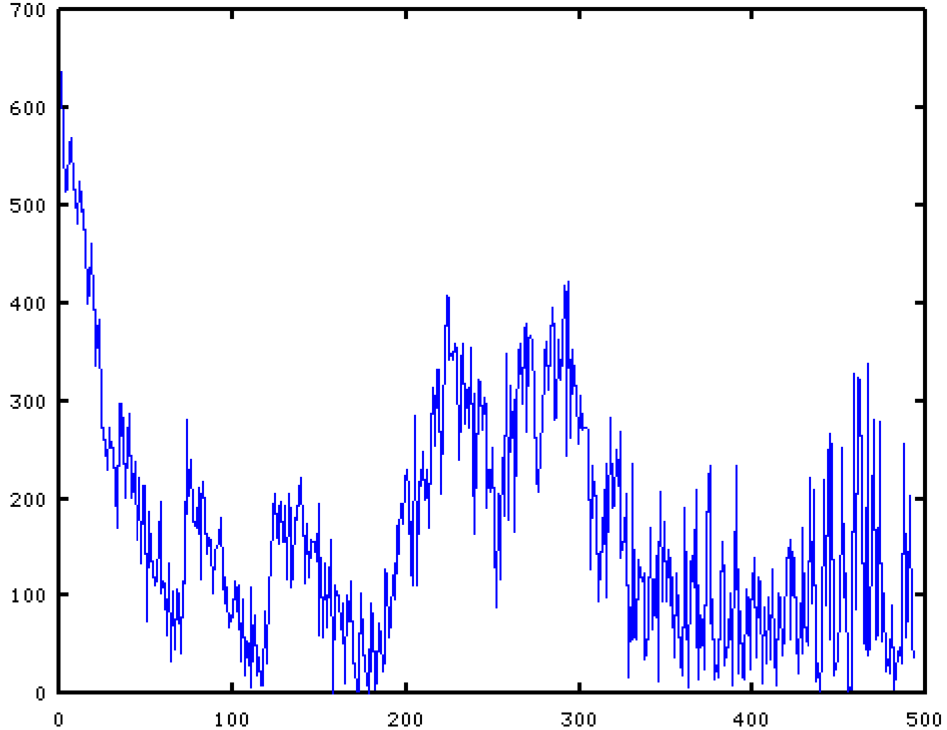
$$\hat{\mathbf{x}} = \left(\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{b}$$

- Iterative least squares solver based on GPS equation
- Solver operates on the partial derivatives of the pseudoranges using Newton's Method
- “Extra station” computed using one station's lat/lon and the aircraft's known altitude – pseudorange changes little with lat/lon changes due to small angular separation

Results

```
Aircraft heard for clock drift estimate: ['abc649', 'a8522b',
'a82aac', 'a98219', 'a6ac39', 'a8a73d']
Total reports used: 640 over 348.59 seconds
drift from 0 relative to station 0: 0.000ppm
drift from 1 relative to station 0: 5.477ppm
mean offset from 0 relative to station 0: 0.000 seconds
mean offset from 1 relative to station 0: 2.219 seconds
```

RMS error in TDOA: 209.9 meters



Better clock drift estimation (filtering instead of mean) would provide significantly reduced error!

Future improvements

- Real-time multilateration instead of post-processing
- Better clock rate estimation
- Kalman filter for position estimation
- CPU optimization on E100 / FPGA work
- Faster sample rates

Questions

- Code is at github.com/bistromath
- Come see me for a live demo on a USRP B100