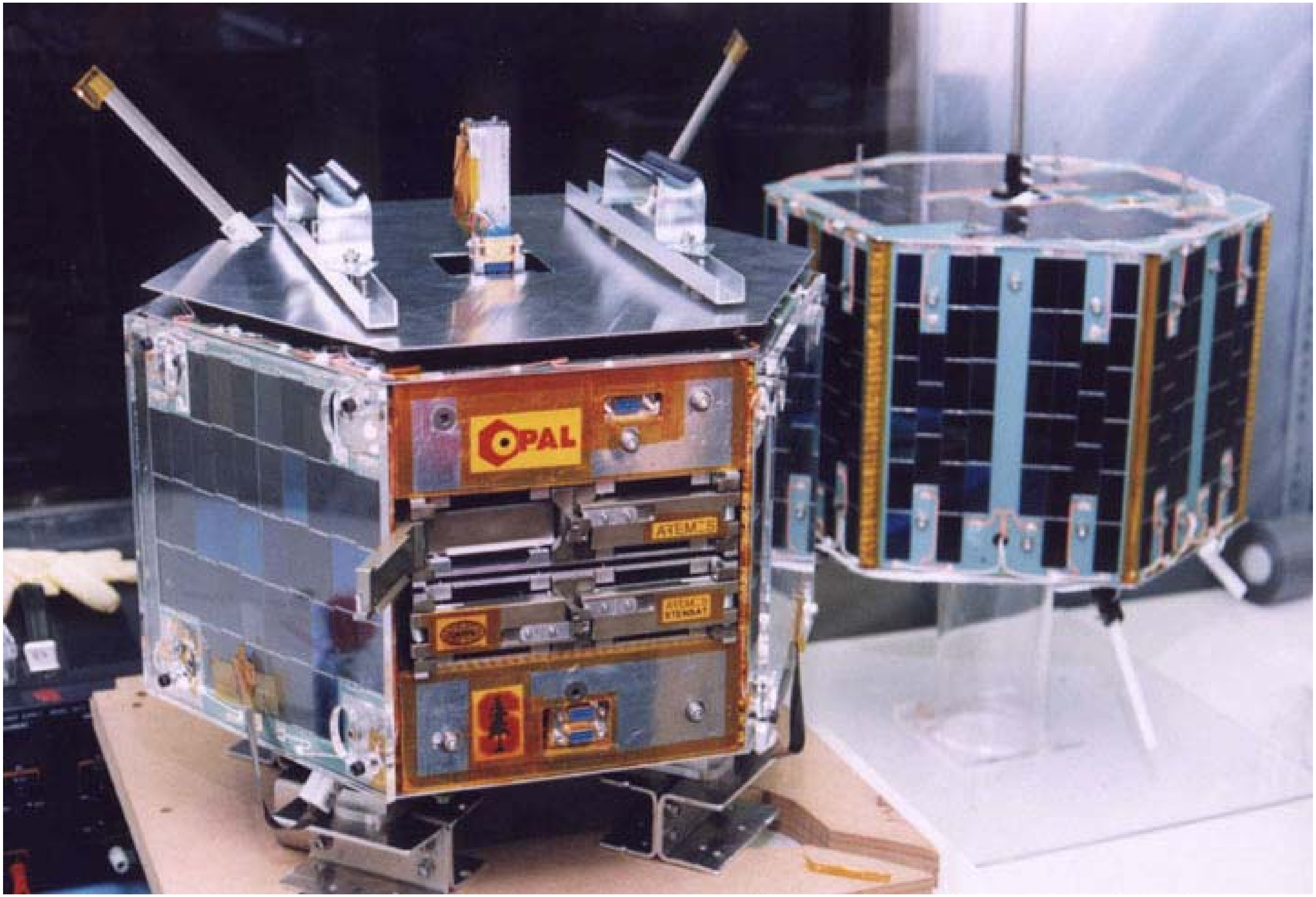
Flexible Application Support in Ground Stations

James Cutler

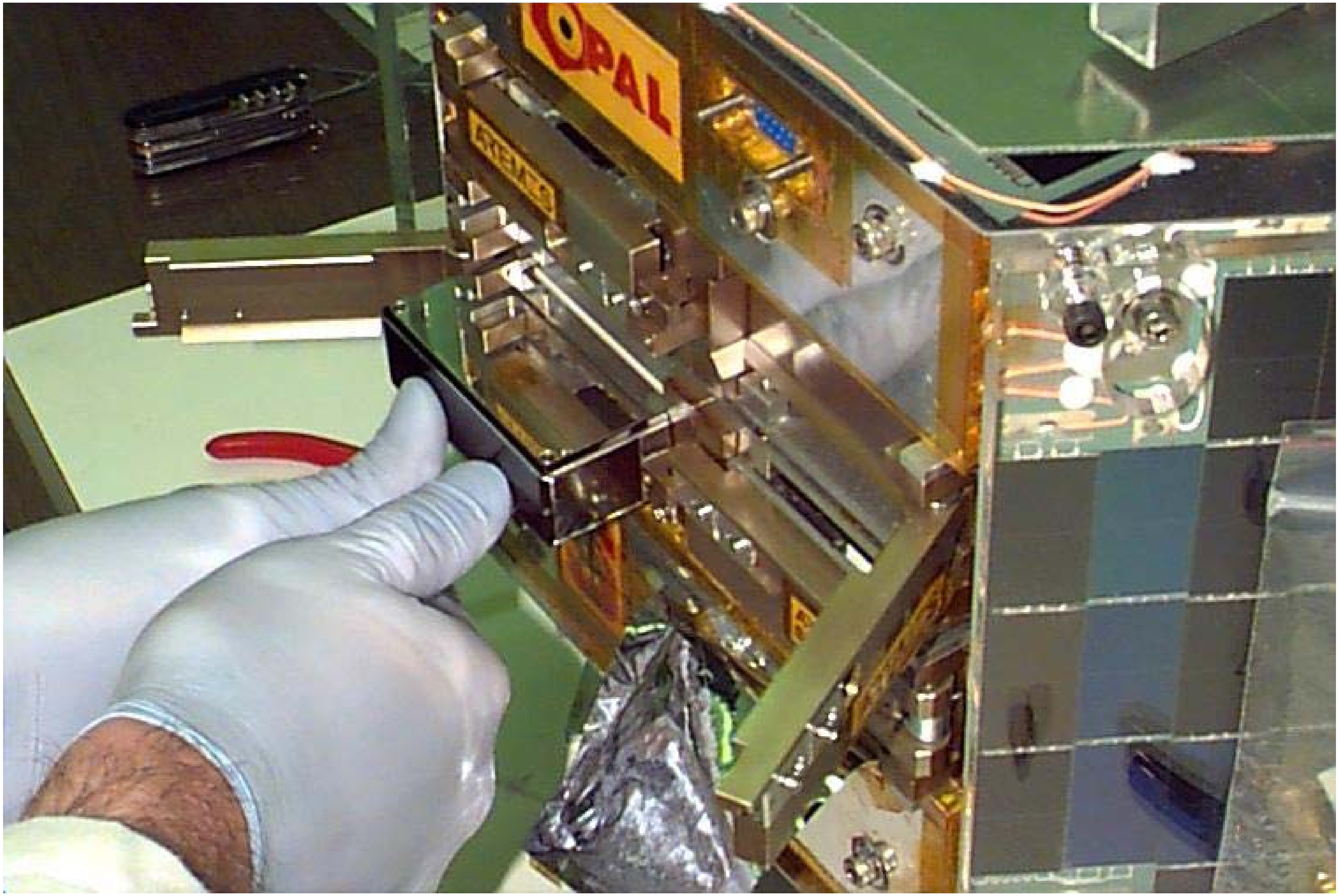
Space Systems Development Laboratory
Software Infrastructures Group
Stanford University

GSAW 2004

Opal and Sapphire



Picosats



Solar Panel Testing

Introduction

- We’re familiar with the trends in space operations.
  - 24x7 connectivity between ground and space.
  - End-to-end access between users and satellites.
  - Some foresee/hope for the end of communication as a constraint.
- With this in mind, the goal of our work is two-fold:
  - Develop infrastructure to make space-based information more accessible.
  - Make this infrastructure robust and reliable while built from unreliable components.
- Primary approach:
  - A network of composable ground stations.
  - High availability through recovery oriented computing.

End-To-End Principles

- To a small degree, the terrestrial networks have reached these goals.
  - The Internet—a network of networks. Heterogeneous devices from cell phones to super computers with applications ranging from instant messaging to stock trading.
  - Past work has been on performance, now shifting towards reliability and total cost of ownership. Recovery-oriented computing (ROC).
- What has enabled grad students in their spare time to create the likes of Yahoo, Google, and Paypal?
  - The Internet is the fertile ground for innovation. What enabled this innovation?
  - Some of have attributed this to the *simplicity of the core Internet* that provides basic building blocks for users to develop applications.
- At the heart of this is the *end-to-end principle*.
  - It states a function can completely and correctly be implemented only with the knowledge and help of the application and the end points.
  - “A lower layer of a system should support the widest possible variety of services and functions, so as to permit unanticipated applications.” [Saltzer, Reed, Clark]

Application to Ground Stations?

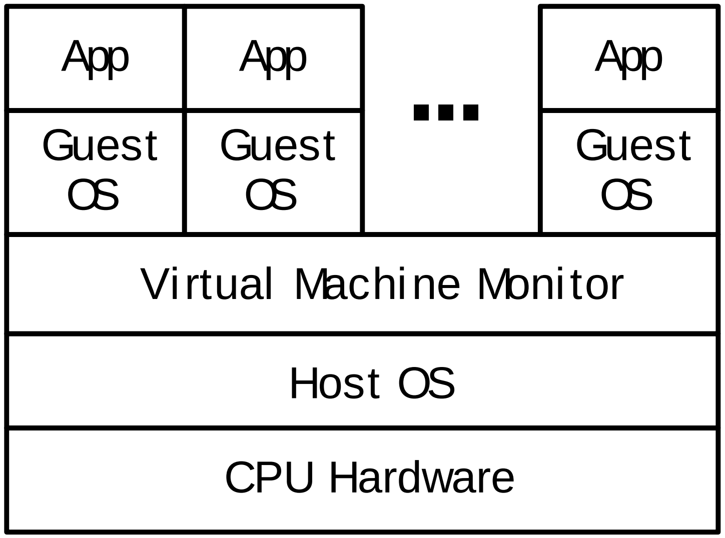
- First, let’s consider computer systems applications of this end-to-end principle:
  - CISC versus RISC—Complex computer architectures that support high level, language specific procedures. Outperformed by reduced instruction set CPUs that could support a variety of languages.
  - SSL—secure socket layer, a protocol that enables negotiation of an encryption technology.
- Now, consider the state of ground stations.
  - They are tackling layers from low level RF to high level application issues such as demultiplexing downlink data.
  - Standardization battles are fought in the ground station and manifest as complexity and high costs.
- This leads to a couple questions:
  - Can we apply the E2E argument to ground stations and provide the same fertile ground for innovation?
  - Can we standardize fundamental ground station services and provide a standard mechanism for *flexible application level support*?

Hierarchical Reference Model

- An attempt to capture core ground station services divided along lines of autonomy into hierarchical layers.
  - Goal is to develop an architecture with core, simple services and a standardized mechanism for flexible application-level services.
- The *virtual hardware* level—captures the fundamental capabilities of low-level ground station components.
  - Master/slave control paradigm where GS exposes low level control of systems.
- The *session* level—captures typical automation tasks and services of a single ground station.
  - Automated tracking, scheduling, health monitoring,etc.
- The *network* level—captures the services of a network of ground stations.
  - Scheduling and GS registry services.
  - Teamed ground stations cooperating on a pass.

digression

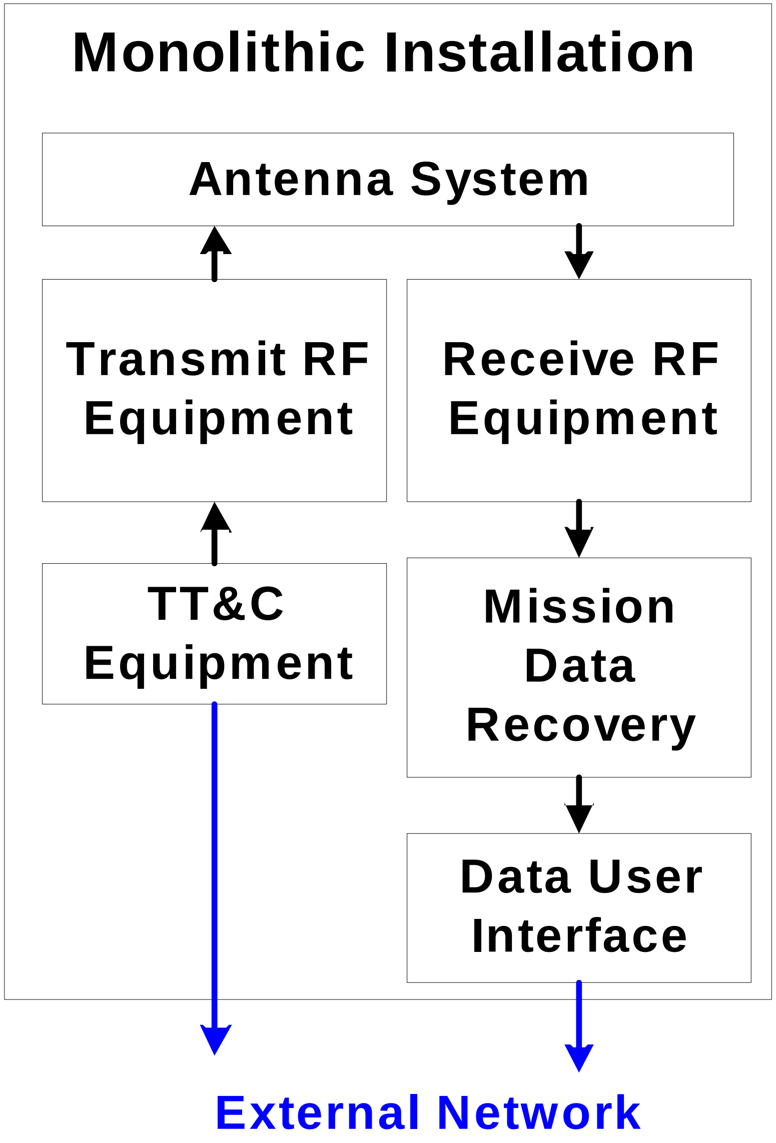
- A host OS and hardware running multiple guest OS, the virtual machines.
  - To the guest, it appears to them as if they are the sole machine.
  - To the host, it just appears as a running. A Virtual Machine monitor (VMM) controls and monitors the VMs. The VM is encapsulated in a file.
  - Common place in IBM main frames for years, but now making their way into mainstream computing (ie Vmware).
- Uses of VMs
  - Guest OS free from the hardware it is running on. Consider HW upgrades now. Just copy.
  - Facilitates backups and restorations.
  - Higher utilization of CPU resources.
  - Isolation, sandboxing, and security.
- What if a core GS service included the ability to run a VM?



Flexible Application Support

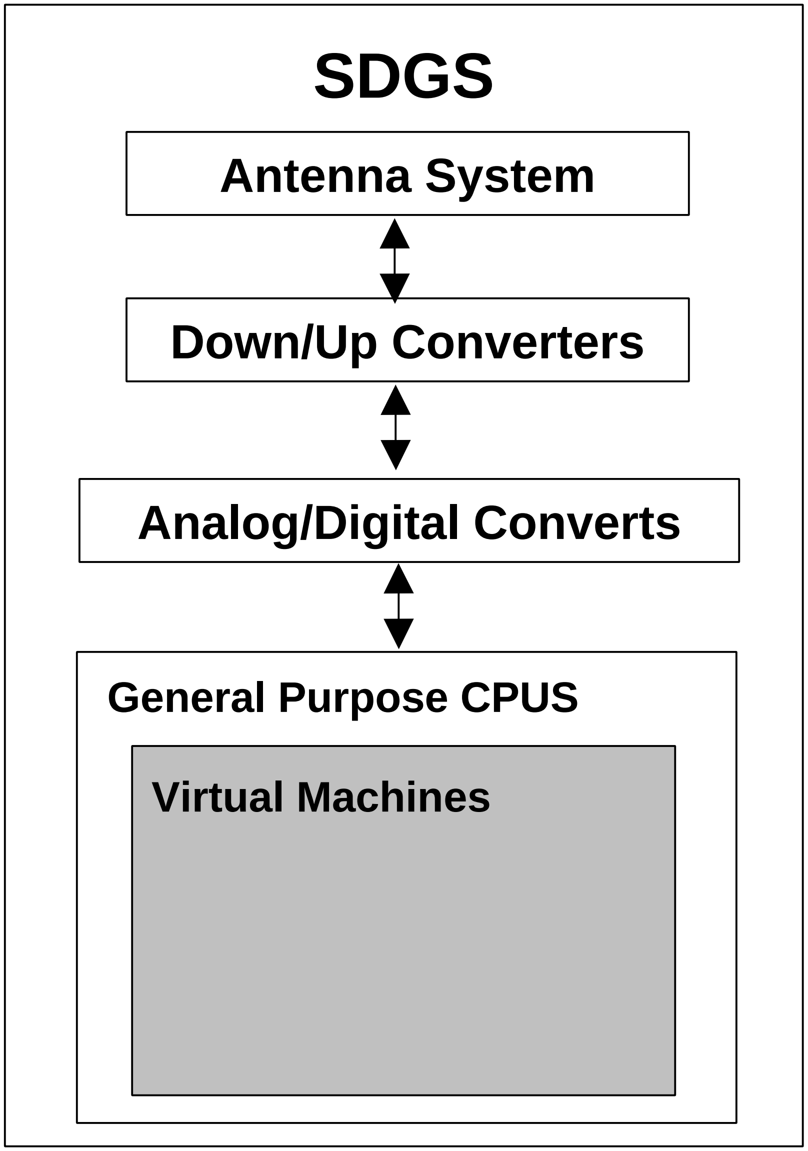
- Due to the nature of ground stations, some application support is likely needed.
  - Custom control of low level hardware devices
  - Large downlinks requiring longer term store and forward.
  - Supporting legacy missions.
- An interesting trend is the move to software intensive systems.
  - Traditionally, distinct hardware devices have been used for bit synchronization, FEC, packetization, security.
  - These services are now being captured in software on general purpose CPUs.
  - A ground station is…an antenna, amplifiers, a ADC, and a CPU.
- Now, combine these software centric ground stations with the concept of a virtual machine.
  - GS users can download their own, custom VM to perform all their bit sync, FEC, packetization, store and forward, data delivery, etc.
  - GS provides a standard mechanism for VM execution. Free from app specific knowledge.

Monolithic Installations



- Traditional, legacy systems.
- Characterized by custom hardware for each of the system components.
- Difficult to upgrade and support multiple missions.
- Fixed components.

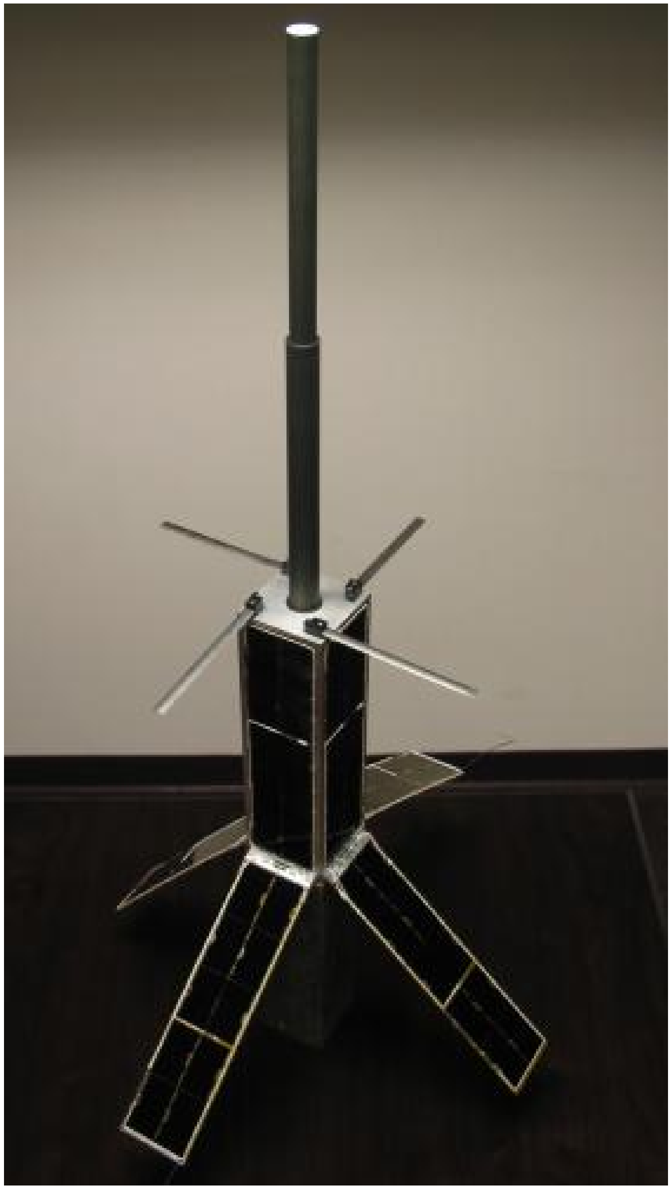
Stations



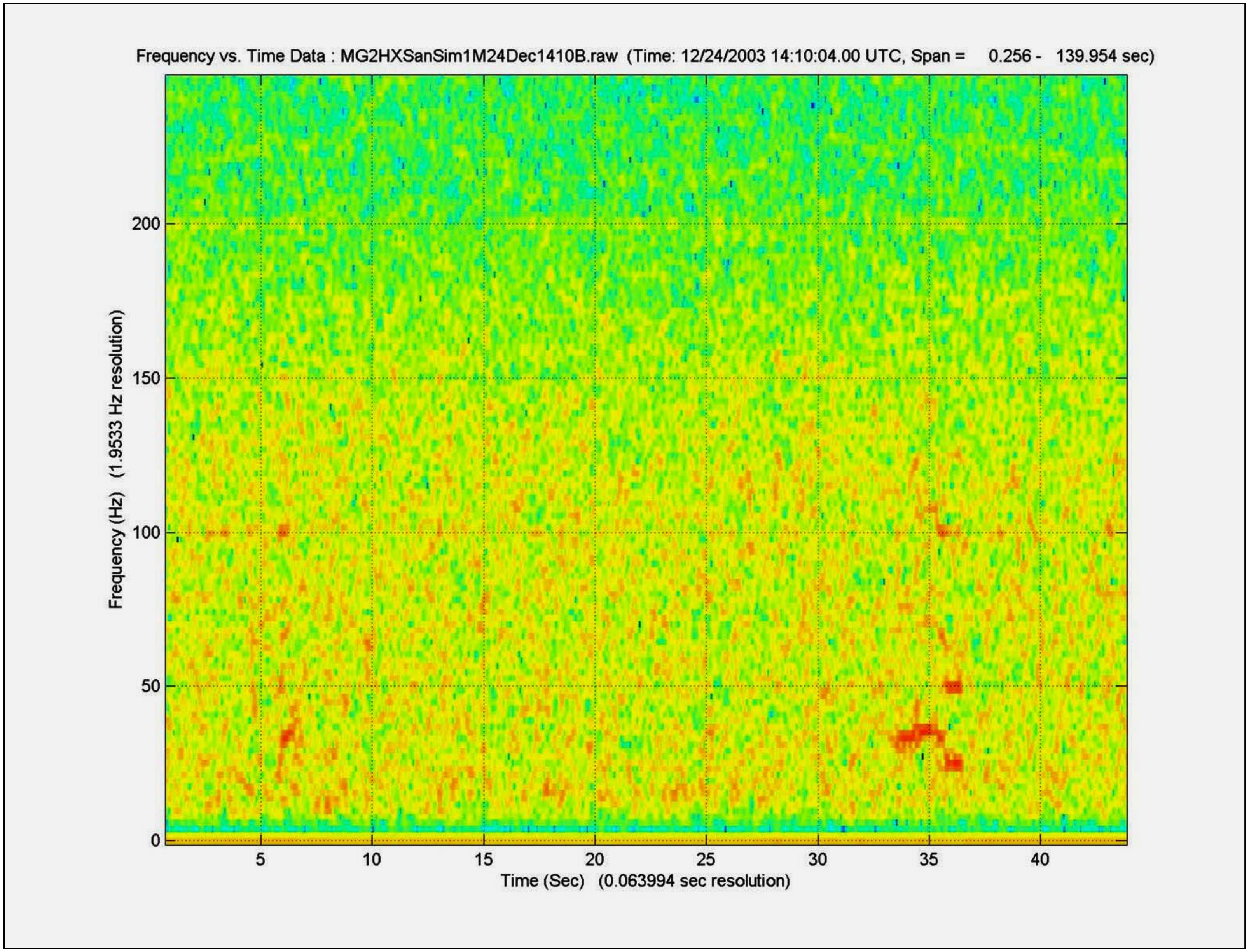
- Reduction in custom hardware
  - Antennas, amplifiers, up/down converters, ADCs, DACs
- Move everything else into a VM
  - Bit sync, FEC, packetization.
  - TTC, misson data, etc.
- VMs are now:
  - Portable
  - Upgradable
  - Customizable
  - Etc.

System

- Developing a reference architecture and implementation—Mercury.
  - Capturing the hierarchical model in an XML-based command and control language—the Ground Station Markup Language (GSML).
  - Enables loosely coupled SW systems and commanding of heterogeneous systems.
  - Built on low cost systems: Linux, Apache, MySQL, PHP, and Java (some C). Open sourced.
  - Upload sat-specific VMs.
- Mercury Ground Station Network (MGSN)
  - Supporting university satellite missions and networking global ground stations.
  - Testbed for operations and Internet accessible.
  - QuakeSat, to the right, supports end-to-end IP. Launched June 30, 2003.



Quake Data


Frequency vs. Time Data : MG2HXSanSim1M24Dec1410B.raw (Time: 12/24/2003 14:10:04.00 UTC, Span = 0.256 - 139.954 sec)
Frequency (Hz) (1.9533 Hz resolution)
0
50
100
150
200
5
10
15
20
25
30
35
40
Time (Sec) (0.063994 sec resolution)

VM Examples

- We’ve developed VM’s for our testbed
  - QuakeSat VM to run application software in Alaska. Fairbanks station plagued by intermittent connectivity during winter.
  - AX.25 and IP over AX.25 VM for use at Stanford.
  - Pacsat VM. Using legacy windows software not internet enabled.
- Other potential VMS
  - Linux router VMs
  - CCSDS Vms
- Other systems using virtual machines
  - PlanetLab—an open, globally distributed platform for developing, deploying, and accessing planetary scale network services.
  - Emulab—another wide area platform for testing and development.
  - Industry support from IBM, Intel, Yahoo, etc. Being deployed to support Internet applications.

Conclusions

- We have discussed the *end-to-end argument* and its application to ground systems.
  - "A lower layer of a system should support the widest possible variety of services and functions, so as to permit unanticipated applications."
- We have developed a reference model that captures core GS services in hierarchical layers based on autonomy.
  - Our reference architecture encapsulates these layers in an XML-based markup language and is built on open-source products.
  - We are deploying this system to support university satellite missions.
- We have proposed a shift in standards work.
  - Instead of focusing on physical to application level standards, let's also standardize a mechanism for flexible application-level support.
  - We are exploring a solution to this based on virtual machine technology.
- Additional information
  - http://swig.stanford.edu/, http://ssdl.stanford.edu/, http://www.mgsn.net/