The SATNet Project

Towards an Open-source Ground Stations Network for CubeSats

authors:
Dr. Ricardo Tubío
Dr. Antonio J. Vázquez
Prof. Jordi Puig
Dr. Naomi Kurahara
Prof. John Bellardo

contact:
rtubiopa@calpoly.edu

04/25/2014
1 State of the Art
  ■ Ground Station Networks
  ■ Current Ground Stations
2 The SATNet Project
  ■ Concept
  ■ Organization
  ■ Release 1
State of the Art

Ground Station Networks
Network Paradigm: client server.

Provides: remote GS operation.

GPLv2 version from 2003 at SourceForge.

Scalability?

Scheduling?
The GSN Network (Japan)

Network Paradigm: **client-server**.
- **GMS**: Ground Station Management Service.
- **GROWS**: GS Remote Operation Web Service.

- Scalability?
- Scheduling?
The MC3 Network (USA)

- Network Paradigm: legacy GCA network.
  - 2 independent servers (scalability?).
  - VPN connection with clients.
- Centralized supervised scheduling.
- Project by the Naval Postgraduate School.
- Propietary software license.
The GENSO Network (Europe)

- Network Paradigm: hybrid peer-to-peer (P2P).
  - AUS: central authentication.
  - MCC/GSS: distributed peers.
- Distributed scheduling.
- Transport through audio transmission.
  - Problems with delay and jitter?
- Network connectivity problems (NAT)?
Current Ground Stations
Current Ground Stations

- Specific per-mission hardware support.
- Not many problems for constructing a new station.
- Networks with a full GS client:
  - SW fits GS → you are in.
  - SW inadequate → you are out.
- Some clients might be an obstacle to access the network.
The SATNet Project

Concept
Network Architecture (1)

- **Network Paradigm:** distributed servers.
  - N-System: main network node.
  - G-Client: ground station client.
  - M-Client: mission operation client.

- Services provided by the central N-System.
- Clients: remote access library.
Each central node creates an isolated sub-network.

Approach similar to IP routers.

Users decide to interconnect nodes for scaling sub-networks.

Nodes interconnection:
- scalability,
- redundancy,
- and privacy.
<table>
<thead>
<tr>
<th>Paradigm</th>
<th>Mercury</th>
<th>GSN</th>
<th>MC3</th>
<th>GENSO</th>
<th>SATNet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remote GS</td>
<td>Client</td>
<td>Client</td>
<td>Legacy</td>
<td>Hybrid</td>
<td>Distributed</td>
</tr>
<tr>
<td>Federation</td>
<td>Server</td>
<td>Server</td>
<td>GCA</td>
<td>P2P¹</td>
<td>Network</td>
</tr>
<tr>
<td>Sources</td>
<td>data</td>
<td>data</td>
<td>data</td>
<td>audio</td>
<td>data²</td>
</tr>
<tr>
<td>License</td>
<td>GPLv2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>Apache v2</td>
</tr>
</tbody>
</table>

1. Initially P2P, finally central server (AUS entity).
2. Custom protocols in between clients.
3. Distributed through the central nodes, similar to a cloud computing approach.
4. Through different subnetworks.
Organization
Development Philosophy

- Open source (Apache V2).
- Incremental software development:
  - 4 software releases.
  - 1 release, 1 subset of functionalities.
- GitHub Project:
  - github.com/satnet-project
- Feel free to join us!
- First node to be hosted at CalPoly.
<table>
<thead>
<tr>
<th>R#</th>
<th>Date</th>
<th>Management</th>
<th>Scheduling</th>
<th>Communications</th>
<th>Additional</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>June 2014</td>
<td>User registration</td>
<td>Basic booking</td>
<td>Assisted</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Basic configuration</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R2</td>
<td>December 2014</td>
<td>Full configuration</td>
<td>Assisted</td>
<td>Non-scheduled</td>
<td>Scalability</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Key distribution</td>
</tr>
<tr>
<td>R3</td>
<td>July 2015</td>
<td>Information service</td>
<td>Private</td>
<td>Private</td>
<td></td>
</tr>
<tr>
<td>R4</td>
<td>September 2015</td>
<td></td>
<td></td>
<td></td>
<td>Testing</td>
</tr>
</tbody>
</table>
Release 1
- **Standalone Node → Private Subnetworks.**
- **WebServices with Django (Python 2.7).**
- **Database: MySQL.**
  - Clustering possibility for redundancy.
- **Communications Service:**
  - Daemon protocol, Twisted over TLS.
  - Better performance for lower delay.
- **Operating system:** Debian Wheezy.
(+) Add Ground Station

Owner: rtublo

STEP 1: Set location

STEP 2: Define your station

Identifier: 

Callsign: 

Elevation (deg): 

Latitude: 35.3479 Longitude: -120.0627
Configuration

- **gs-calpoly**

  - **Radio Callsign:** gscalaip900
  - **Contact elevation (deg):** 10.50
  - **Coordinates:** (35.3804, -120.4565)

Communication Channels

- **u-fm-1**
  - **IARU band:** 70 cm
  - **AMSAT letter:** U
  - **(435.000000, 438.000000)**
  - **Modulation(s):** FM
  - **Bitrate(s):** 600, 900
  - **Bandwidth(s):** 12,500
  - **Polarization(s):** Any

- **ch-fm-2**
  - **IARU band:** 70 cm
  - **AMSAT letter:** U
  - **(435.000000, 438.000000)**
  - **Modulation(s):** FM
  - **Bitrate(s):** 300, 900, 1200
  - **Bandwidth(s):** 25,000
  - **Polarization(s):** Any

Channels Schedule

*(no rules for ch-fm-2)*
Thanks for your attention!