A Testbed for Demonstration and Performance Analysis of an Autonomous Scheduling System for Communications Nanosatellites

Peter J. Yoo
Alonzo E. Jenkins
Cherry Y. Wakayama
Mohammad R. Alam

SPAWAR Systems Center Pacific
San Diego, CA
Need: Mission-level Hardware testbed

- Software-in-the-loop (SITL) modeling and simulation has limitations
- No low-cost Hardware-in-the-loop (HITL) M&S tool to test/simulate mission scenarios
- Flatsats / ground testbeds can only simulate individual nanosatellites
Proposed Solution

▼ Create a low-cost, mission-level simulation environment using Commercial-Off-The-Shelf (COTS) hardware
- Hardware-In-The-Loop (HITL) nanosatellite constellation testbed
- Open source, industry standards and existing flight software
- Graphical User Interface (GUI) for input parameters and analysis output

▼ Demonstrate autonomous algorithms using the testbed
Virtual Ground Node – Simulates ground stations in different geographic areas and hosts GUI and module for automation and optimization algorithms

Orbit Simulator – Provides orbit information for a given mission

Flight SW Core – Provide services for sending commands / receiving telemetry

Orbit Propagator – propagate position and velocity of a satellite

Subsystem Simulator – Simulates battery, solar panel, attitude control, gps receiver

Algorithm Module – User-defined automation and optimization algorithms
Nanosat Communications Constellation Testbed Design

▼ Hardware Implementation

- 1 x Workstation
- 4 x BeagleBone Black (BBB)
- 1 x Ethernet switch (8-port)

▼ Software Implementation

- Flight Software developed for DoD 6U nanosat bus
- VMware with Ubuntu
- Python IDE

*https://beagleboard.org
Flight SW Core Description

- Testbed uses actual flight software
- Uses Consultative Committee for Space Data Systems (CCSDS) compliant message structure called “Space Packet” for all telemetry, commands, interface control requests and acknowledgements

![Space Packet Diagram]

- IP-based/Ethernet interface between components including ground stations
- 9 Services* available between bus and payload including
  - Payload Command Forwarding
  - Payload Telemetry
  - Bus Command
  - Bus Telemetry Packet
  - Bus Telemetry Stream
  - Payload Data Storage
  - Payload Data Downlink
  - Payload Data Load
  - Time Service

* SN-BPLICD-001 Rev 1.0 by Naval Research Laboratory
Scenario Simulation: Graphical User Interface Design Example

1. Select source ground destination ground node
2. Generate message (priority, message type, size, etc.)
3. Check satellite access window
4. Check result for analysis
Nanosat Communications Constellation
Testbed Message flow

Scenario Simulation

Start

Choose message origin (ground node)

Choose message destination (ground node)

Yes

Populate satellite access table

User defined scheduling algorithm (Choose satellite)

Is satellite visible to origin?

Yes

Generate Message

Yes

Store message

Propagate orbit

No

Entire message uplinked?

Yes

Uplink Message to satellite

No

Queue message and wait for next pass

No

Entire message downlinked?

Yes

Downlink Message based on algorithm

No

Queue message and wait for next pass

No (Wait)

Destination Ground Node

Run analysis

Store message

Yes

End

Origin Ground Node

Entire message uplinked?

Yes

Queue message and wait for next pass

No

Is satellite visible to destination?

Yes

Run User defined Scheduling algorithm

No

No (Wait)

Nanosatellite

User defined scheduling algorithm (Choose satellite)

Is satellite visible to origin?

Yes

Generate Message

Store message

Propagate orbit

No

Entire message downlinked?

Yes

Downlink Message based on algorithm

No

Queue message and wait for next pass

No (Wait)
Interface to Algorithm Module

- Software wrapper allows rapid implementation of user-defined scheduling algorithms
  - Provides receive and send capability between ground nodes and nanosatellites
  - Telemetry data is available for algorithm to use during decision making

- Algorithm Module provides pathway to implement optimization in the flight computer
  - Communication scheduling
  - Other event scheduling
Summary

- Developed mission-level nanosatellite communications constellation testbed framework using COTS components
- Defined virtual ground node and flight computer module interfaces
- Developed a GUI design to help modeling and simulation
- Developed message delivery simulation test cases for evaluation

Ongoing Work

- Model a nanosat subsystem simulation module (battery, solar panel and etc.)
- Implement energy-cognizant nanosat message delivery scheduling system
- Implement MC3 ground station network model and simulation
- Demonstrate autonomous message delivery scheduling systems for a store-and-forward nanosat communication constellation using the testbed
THANK YOU