Initial Results from ACCESS: an Autonomous CubeSat Constellation Scheduling System for Earth Observation

Andrew “Kit” Kennedy, Prof. Kerri Cahoy
akennedy@mit.edu
August 8, 2017

SmallSat Session V - Ground Systems
Outline

• Motivation
• Approach
  – EO Constellation Scheduling
  – ACCESS architecture
  – Data Routing
  – Simulation cases
• Results
  – Data Routing latency
  – Urgent Data Routing latency
  – Execution Time
• Conclusion
Outline

• Motivation
• Approach
  – EO Constellation Scheduling
  – ACCESS architecture
  – Data Routing
  – Simulation cases
• Results
  – Data Routing latency
  – Urgent Data Routing latency
  – Execution Time
• Conclusion
EO Constellation Scheduling

• Existing tools: observation and downlink scheduling
  – Planet Inc. algorithms
  – Multi-Sat Multi-GS scheduling
  – STK Scheduler

• Crosslink usage with tight-knit satellite clusters
  – Task allocation (e.g. market based)
  – Local or mesh networks
1. Simulate a “spread-out” satellite constellation
2. Schedule with a centralized ground planning system
   – Key: utilize long-distance crosslinks for low-latency bulk data routing
3. Distribute plans to sats via ground and crosslink network
4. Reactive observation replanning onboard sats
   – Key: distribute updates through network
Data Routing Approach

- Optimize metric: observation latency to downlink
- Implemented a greedy algorithm
  - Downlink observations in temporal order
  - Use earliest downlink possible each time
Data Routing Approach

- Optimize metric: observation latency to downlink
- Implemented a greedy algorithm
  - Downlink observations in temporal order
  - Use earliest downlink possible each time

- Downlink up to X Mbit of Obs1
- Add Obs1 back to queue
- Go to next observation
Payload and Link Models

- **High data rate EO payload**
  - 5 spectral bands, optical and NIR
  - 127.5 Mbps compressed data
  - 60 s average flyover

- **X-band Downlink**
  - 1 W Tx
  - 0.25U
  - 5.5 m Rx diam.
  - Adaptive data rate
  - 25-45 Mbps

- **Optical Crosslink**
  - 1 W Tx
  - 1U
  - 8.5 cm Rx diam.
  - Adaptive data rate
  - 10 Mbps @ 4,300km range
Simulation Cases

- 24h window for routing
- Set of 33 obs. targets
- 3 orbital geometries
- 3 GS networks
Outline

- Motivation
- Approach
  - EO Constellation Scheduling
  - ACCESS architecture
  - Data Routing
  - Simulation cases
- Results
  - Data Routing latency
  - Urgent Data Routing latency
  - Execution Time
- Conclusion
Routing Latency

- Routing Latency results
  - For first 1 Gbit of data from each observation
  - Average of latencies for all observed data packets
- Do not yet consider satellite energy constraints

Example

Downlink only: Blue

Downlink + Crosslinks: Red

150.0
145.0
9 GS
Routing Latency: 10 Sat SSO

- 10 satellites in single 10:30 LTAN SSO

Latency improves with more GS

Average Route Latency (minutes)

- 1 GS: 121.1 minutes
- 9 GS: 52.5 minutes
- 17 GS: 37.9 minutes

Xlnks reduce latency 50% or more. Latency < 1h: “desirable”
Better Latency: 30 Sat Walker

- 30 satellites in a 3 plane Walker Delta pattern, 60° inc.

Xlnks reduce latency ~80% or more. Latency < 0.5h: closing in on instantaneous

See a large latency increase in downlink-only case (121 min for 10 sats SSO)
Urgent Data Routing

- Same 33 targets
- Subset of targets designated “urgent” for ~2 h durations
  - Downlinked before all other obs
  - Simulates changing observation priorities
Urgent Latency, Downlink Only

- Plot with downlinks only, 9 GS

Urgent latency slowly degrades as more marked urgent

Latency reduced more than half when marked urgent
Xlinks Reduce Urgent Latency

- Plot with downlinks and crosslinks, 9 GS

With xlnks, latency driven even lower for urgent obs
23.3 mins to 15.9 mins
Data Routing Execution Time

- Measured algorithm execution time
  - Scheduling of obs, dlinks, xlinks; data packet routing
  - Custom Python code
- For increasing constellation size
- For two planning window durations: 12 hours and 24 hours
- Run on a 2013 Macbook Pro laptop (2 GHz, 8 GB RAM)

<table>
<thead>
<tr>
<th>Number of Satellites</th>
<th>Execution Time (mins)</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>12 Hour Window</td>
<td>24 Hour Window</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>0.18</td>
<td>0.56</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>0.94</td>
<td>2.92</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>4.57</td>
<td>13.23</td>
<td></td>
</tr>
</tbody>
</table>

Planning execution time appears tractable for scalability.
Outline

• Motivation
• Approach
  – EO Constellation Scheduling
  – ACCESS architecture
  – Data Routing
  – Simulation cases
• Results
  – Data Routing latency
  – Urgent Data Routing latency
  – Execution Time
• Conclusion
Conclusion

• Summary of results
  – Regular latency, Walker Delta with xlnks: **23 and 17 mins**
  – Urgent latency, Walker Delta with xlnks: **16 mins**
  – **Execution time of 13 mins** for 24 h window with 100 sats

• Long range crosslinks promising for low latency bulk data delivery

• Future work
  – **Algorithm improvements**: energy-aware planning, data routing optimization (utilizing e.g. MILP), onboard replanning.
  – **Additional metrics, sensitivity studies** (particularly: crosslink range and data rate, simplex vs duplex)
  – **Incorporation in operations SW stack**
  – Open-sourcing
References

• **Slide 5**

  10. [https://www.nasa.gov/directorates/spacetech/small_spacecraft/edsn.html](https://www.nasa.gov/directorates/spacetech/small_spacecraft/edsn.html)

• **Slide 9**

Backup
The Problem: EO Data Delivery

• To effectively monitor events on Earth, we need “almost instantaneous data availability” \(^3,4\)
  – 0.5 to 1 hour \(^5\)
  – Benchmark: 90min latency, Disaster Monitoring Constellation \(^3\)

Floods

Eruptions, Fires

Earthquakes

Zhang et al. \(^6\)

Pergola et al. \(^7\)

Liu et al. \(^8\)

and more…
• Inter-satellite crosslinks
  – TLM and CMD
  – Bulk data routing
• Crosslinks stress operations
  – Energy usage
  – Satellite scheduling complexity
  – Constellation scheduling complexity
Imaging Payload Details

• From commercial 6U CubeSat design by Tsitas and Kingston [x21]
  – Designed to be competitive with DMC and RapidEye EO satellites
  – 600 km SSO, GSD of 6.5m and swath width of 26km

• Imager
  – Questar 3.5 telescope (89mm aperture, 20.3cm length, 1.4 kg)
  – Fairchild imaging CCD5061 (4000 pixels, 12 bit digitization)

• 5 spectral bands, 255 Mbps uncompressed
  – 2:1 lossless compression -> 127.5 Mbps
Example Plot of Battery Level
Payload/Comm Energy Usage

Clements et al, 2016

Chart showing orbital energy usage for various payloads and communication methods, with capacities and power consumption listed for different data transfer rates.
Motivation

• What if there were a **low cost** way for a CubeSat to downlink 100 Gb/day?
  – Most CubeSats downlink $<< 10$ Gb/day (UHF or S-band systems) \(^{[1]}\)

• Radio frequency (RF) downlinks challenged by resource constraints
  – E.g., ground station size, transmitter power, or spectrum

• **Lasercom is less resource constrained and could scale to Gbps** \(^{[3]}\)
  – More power-efficient for given size, weight, and power (SWaP)
  – More bandwidth available
  – Many groups working on it: MIT, Aerospace Corporation, Sinclair, UF, DLR, JAXA, …

![Wallop's CubeSat Comm. Antennas\(^{[2]}\)](UHF, 18.3 m S band, 11 m)

![MIT Lasercom Ground Station](MIT Lasercom Ground Station)
## NODE Space Terminal Overview

<table>
<thead>
<tr>
<th>Scope</th>
<th>CubeSat Low-Cost Payload (&lt;$15k parts)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architecture</td>
<td>Direct detection MOPA</td>
</tr>
<tr>
<td></td>
<td>COTS telecom parts (1550 nm)</td>
</tr>
<tr>
<td>Downlink data rates</td>
<td>10 Mbps (30 cm amateur telescope)</td>
</tr>
<tr>
<td></td>
<td>100 Mbps (1 m OCTL)</td>
</tr>
<tr>
<td>Power</td>
<td>0.2 W (transmit power), 15 W (consumed power)</td>
</tr>
<tr>
<td>Beamwidth</td>
<td>1.3 mrad half power (initial demo)</td>
</tr>
<tr>
<td>Modulation</td>
<td>PPM</td>
</tr>
<tr>
<td>Coding</td>
<td>RS(255,239)</td>
</tr>
<tr>
<td>Mass, Vol.</td>
<td>1.0 kg, 1 U</td>
</tr>
</tbody>
</table>

### Control architecture
- Bus coarse pointing (<0.5°)
- FSM fine steering (+/- 2.5°)
- Beacon receiver (976 nm) for pointing knowledge (20 arcsec)

### Current Status
- Pointing control testing
- Component-level environmental tests
- Functional testing
- End-to-end over the air demo
Future Work

- Deployment of global planner algorithm on ground software stack (e.g. Ball Aerospace’s COSMOS)
- Deployment local (satellite) planner algorithm on flight software stack (e.g. NASA Goddard’s cFS)
- Incorporate more versatile observation payload and satellite operations modeling
- Open source release of ACCESS software for use by the wider small sat community.
Earth Observing Constellations

• Advantages
  – Higher temporal resolution
  – Multi-point instrument coordination
  – Low-latency data availability

TROPICS Mission, MIT LL
Ground Stations

- Expensive to deploy
- Lots of organizational/legal overhead
- Very hard to deploy across oceans
- For lasercomm, clouds can hinder downlink
- For commercial networks
  - Still have to pay for usage
  - Have to worry about schedule access
Latency: Both Geometries

- Combined latency plot of 10 sat SSO, 30 sat Walker

Dlnk latency increased for larger constellation! Due to skew in downlink latency.

Xlnk latency significantly lower with a larger constellation.

<table>
<thead>
<tr>
<th>Average Route Latency (minutes)</th>
<th>10s: No Xlnks</th>
<th>10s: Xlnks</th>
<th>30s: No Xlnks</th>
<th>30s: Xlnks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wallops</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BridgeSat</td>
<td>52.5</td>
<td>23.3</td>
<td>121.1</td>
<td></td>
</tr>
<tr>
<td>SFN</td>
<td>37.9</td>
<td>17.2</td>
<td>79.9</td>
<td>78.8</td>
</tr>
</tbody>
</table>
More revisits, lower latency

• What we need:
  - Lower inter-revisit times to targets
  - Less time from data collection to delivery

• How we get there
  - Larger constellations
  - More ground stations
  - Inter-satellite crosslinks
  - More frequent flyovers of targets
  - Lower wait time for downlink
  - More total volume to ground
  - Route data to downlinks
  - Distribute bandwidth over ground stations
Scaling Operations

Need an automated operations approach that:

- Scales to many satellites (tens to hundreds)
- Efficiently balances data collection and routing
- Handles unique constraints of CubeSat platform

- Human-in-the-loop planning scales linearly with number of satellites [x3]
- EO Data rates of 100 MB to TB per orbit [x2,x4,x5]
- Often impossible to fully downlink all data
- Limited comm. availability
- Low energy generation, storage
- Multi-modal measurements
ACCESS Design Goals

• Efficiently manage data collection and routing to ground
  – Schedule observations, downlinks, and crosslink to balance fast
downlink of key data with bulk data delivery
  – “efficient” – not optimal scheduling, but close enough
  – Key advantage: crosslink routing built directly into algorithms

• Allow scalability to 100s of satellites
  – Scheduling divided based on constellation-level and satellite-level
constraints
  – Sacrifices some degree of optimality in scheduling for better
tractability

• Enable reactive and federated constellation operations
  – Satellites have some freedom to replan activities
  – Allows reactivity for disaster monitoring, multi-constellation
cooperation
  – Key advantage: loose coupling of planning responsibility
between ground and satellites
ACCESS CubeSat Ops Model

- 3 activities
  - Observation
  - Crosslink
  - Downlink

- Power usage for activities added on base-level (“idle”)
ACCESS Architecture

L1 Global Planner

Activity timings

L2 Local Planner

Flight Software

Satellite 1

Satellite 2

Satellite n

Ground. Considers:
- Data collection
- Data routing through xlink, dlink

Activity timings, weightings

Detailed Dynamics Simulator

Satellite. Considers:
- Current sat state
- New observation opportunities

Orbit and Communications Forecaster

Telemetry and Command Manager

CMD

TLM

state data
Background: Scheduling

- Algorithms and software exist for small satellite scheduling
  - Manage activity timing and limited onboard resources
  - e.g. Planet Inc. [x8], Multi-Sat Multi-GS scheduling [x9], ASPEN/CASPER [x10], STK Scheduler [x6,x7]
- EO constellation management adds difficult logistics
  - Tasking satellites with observation targets [x8]
  - De-conflicting downlinks between satellites [x8,x9]
  - Maintaining schedule synchronization across constellation [x11,x12,x13,x14,x15]
- Using crosslinks as data routes add more complexity
  - At first glance, number of connections between satellites grows as $N^2$