Autonomous Scheduling of Agile Spacecraft Constellations with Delay Tolerant Networking for Reactive Imaging

Sreeja Nag[1], Alan S. Li[1], Vinay Ravindra[1], Marc Sanchez Net[2], Kar-Ming Cheung[2], Rod Lammers[3], and **Brian Bledsoe[3]**

[1]NASA Ames Research Center, Bay Area Environmental Research Institute, CA, USA
[2]Jet Propulsion Laboratory, California Institute of Technology, CA, USA
[3]University of Georgia, GA, USA
Sreeja.Nag@nasa.gov

Abstract

Small spacecraft now have precise attitude control systems available commercially, allowing them to slew in 3 degrees of freedom, and capture images within short notice. When combined with appropriate software, this agility can significantly increase response rate, revisit time and coverage. In prior work, we have demonstrated an algorithmic framework that combines orbital mechanics, attitude control and scheduling optimization to plan the time-varying, full-body orientation of agile, small spacecraft in a constellation. The proposed schedule optimization would run at the ground station autonomously, and the resultant schedules uplinked to the spacecraft for execution. The algorithm is generalizable over small steerable spacecraft, control capability, sensor specs, imaging requirements, and regions of interest. In this article, we modify the algorithm to run onboard small spacecraft, such that the constellation can make time-sensitive decisions to slew and capture images autonomously, without ground control. We have developed a communication module based on Delay/Disruption Tolerant Networking (DTN) for onboard data management and routing among the satellites, which will work in conjunction with the other modules to optimize the schedule of agile communication and steering. We then apply this preliminary framework on representative constellations to simulate targeted measurements of episodic precipitation events and subsequent urban floods. The command and control efficiency of our agile algorithm is compared to non-agile (11.3x improvement) and non-DTN (16% improvement) constellations.

Introduction

Response and revisit requirements for Earth Observation (EO) vary significantly by application, ranging from less than an hour to monitor disasters, to daily for meteorology, to weekly for land cover monitoring (Sandau, Roeser, and Valenzuela 2010). Geostationary satellites provide frequent revisits, but at the cost of coarse spatial resolution, extra launch costs and no polar access. Lower Earth Orbit satellites overcome these shortcomings, but need numbers and coordination to make up for response characteristics. Adding agility to satellites and autonomy to the constellation improves the revisit/response for the same number of satellites in given orbits. However, human operators are expected to scale linearly with constellation nodes (Eickhoff 2011) and operations staffing may be very costly.

Copyright © 2019. All rights reserved.

Earth-Observing Constellation Autonomy

Scheduling algorithms for agile EO have been successfully developed for single large satellite missions, examples being Automated Scheduling and Planning Environment for EO-1 (Sherwood et al. 1998), scheduling for the ASTER Radiometer (Muraoka et al. 1998) on the Terra satellite, high resolution imagery from the IKONOS commercial satellite (Martin 2002), scheduling observations for the geostationary GEO-CAPE satellite (Frank, Do, and Tran 2016) and scheduling image strips over Taiwan by ROCSAT-II (Lin et al. 2005). The hyperspectral instrument CHRIS, carried on the Proba spacecraft, demonstrated dynamic pointing for multi-angle imaging of specific ground spots that it is commanded to observe (Barnsley et al. 2004). Scheduling 3-DOF observations for large satellite constellations has been formulated for the French PLEIADES project (Lemaître et al. 2002; Damiani, Verfaillie, and Charmeau 2005) and COSMO-SkyMed constellation of synthetic aperture radars (Bianchessi and Righini 2008). Scheduling simulations have demonstrated single Cubesat downlink to a network of ground stations within available storage, energy and time constraints.

Recent advances in small and agile satellite technology have spurred literature on scheduling fleet operations. Coordinated planners in simulation (Abramson et al. 2013; Robinson et al. 2017) can handle a continuous stream of image requests from users, by finding opportunities of collection and scheduling air or space assets. Cubesat constellation studies (Cahoy and Kennedy 2017) have successfully scheduled downlink for a fleet, aided by inter-sat communication. Evolutionary algorithms for single spacecraft (Xhafa et al. 2012), multiple payloads (Jian and Cheng

2008) and satellite fleets (Globus et al. 2002) are very accurate but at large computational cost due to their sensitivity to initial condition dependence (genetic algorithms), exponential time to converge (simulated annealing) or large training sets (neural nets). Agile constellation scheduling with slew-time variations have shown reasonable convergence in the recent past using hierarchical division of assignment (He et al. 2019) and step-and-stare approaches using matrix imagers (Shao et al. 2018). However, the published algorithms have not been developed for onboard execution on real-time, fast-response EO applications and do not consider inter-sat comm scheduling in conjunction with imaging operations.

We have recently demonstrated (Nag, Li, and Merrick 2017) a ground-based, autonomous scheduling algorithm that optimizes spacecraft attitude control systems (ACS) to maximize collected images and/or imaging time, given a known constellation. The algorithm is mow broadened in application scope by leveraging inter-satellite links and onboard processing of images for intelligent decision-making. Improving coordination among multiple spacecraft allows for faster response to changing environments, at the cost of increased scheduling complexity.

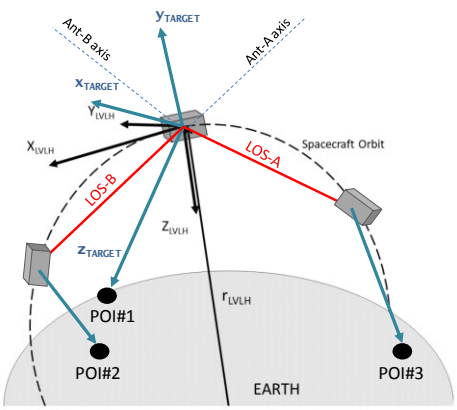

Figure 1—A constellation of satellites observing three points of interest (POI) by agile steering of their body frames, based on information shared when they have line of sight (LOS).

Networked Constellations and Reactive Science

DARPA's delay/disruption tolerant network or DTN paradigm (Cerf et al. 2007) is an emerging protocol standard for routing in the dynamic and intermittent operation environment. DTN makes it possible to minimize replication and improve the delivery probability within available resources, but has never been applied to EO inter-sat data exchange. We show that DTN enabled, agile constellations can respond to transient, episodic and/or extreme events using an autonomous scheduling algorithm that is executable onboard. The target scenarios are simulated by a simplified Observing System Simulation Experiment (OSSE) to evaluate the benefit of our proposed algorithm.

In the traditional sense, an OSSE is a data analysis experiment used to evaluate the impact of new observing systems, e.g. satellite instruments, on operational forecasts when actual observational data are not fully available (Arnold and Dey, 1986). An OSSE comprises of a free-running model used as the ground truth ('nature run'), used to compute the 'synthetic observations' for any observing systems, with added random errors representative of measurement uncertainty. Synthetic observations represent a small, noisy subset of the ground truth. They are used to forecast the full ground truth, then compared with the nature run. The disparity between the nature run of a chosen scenario, and the instrument-derived forecasts is then used to inform better instrument or mission design (Feldman et al. 2011). OSSEs can be used for mission planning, because different operational options can be assessed for different relevancy scenarios by changing the observing system characteristics and nature run appropriately (Nag, Gatebe, and Weck 2015; Nag et al. 2016)..

Methodology

We propose a novel algorithmic framework that combines physical models of orbital mechanics (OM), attitude control systems (ACS), and inter-satellite communications and optimizes the schedule for any satellite in a constellation to observe a known set of ground regions with rapidly changing parameters and observation requirements. The proposed algorithm is expected to run on the satellites, so that they can make observation decisions in a distributed fashion and react to the changing ground conditions as quickly as possible. The algorithm is flexible, and can run on the ground, i.e. the satellites can downlink their observed data, the ground will run the proposed algorithms, and uplink the resultant schedule to the satellite. Since the ground stations are expected to be inter-connected on Earth and perfectly in sync with each other at all times, the optimization is centralized and the resultant schedule avoids the problem of redundant observations due to lack of consensus among the satellites. A ground run will also significantly reduce the onboard processing requirements on the satellites. However, since information exchange occurs at only sat-ground contacts, the computations would be based on significantly staler data compared to the distributed, onboard process. In a rapidly changing environment, an hour or two of latency in reported measurements of physical phenomena reduces the effectiveness of our proposed algorithm, which can otherwise ingest the recomputed value of observations in any region, based on recent observations.

The algorithmic framework is described briefly, and its corresponding information flow summarized in Figure 1. The OM module propagates orbits in a known constellation and computes possible coverage of known regions of

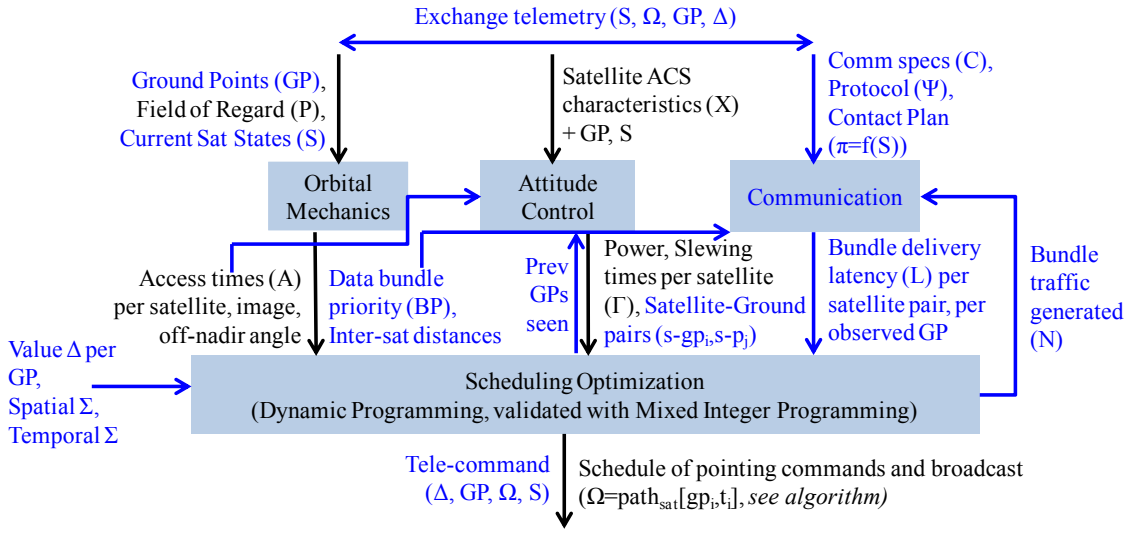

Figure 2 – Major information flows between the modules in the proposed agile EO scheduler, expected to run onboard every satellite in a given constellation, applied to global urban flooding in this paper. This framework can exchange information (as identified at the top) between the satellites via peer-to-peer communication or via the ground (reverse bent pipe architecture). The blue arrows/text represent newly added components to the previous version of the algorithmic framework (Nag, Li, and Merrick 2017).

interest (appropriately discretized into ground points – GP) within a field of regard (FOR). It makes available the propagated states and possible access times per satellite and per GP to the ACS. The ACS uses this information, with known satellite and subsystem (hardware) characteristics to compute: the time required to by any satellite at a given time to slew from one GP to another (including satellite movement), resultant power, momentum and stabilization profiles. Based on the ordering of satellites expected to access a region after any given satellite, the OM also makes available line-of-sight (LOS) availability and corresponding inter-sat distances in the constellation at any time, as well as the priority of bundle delivery to the Comm module. Any DTN implementation, even if it works via bundle routing through any broadcastable satellite, then knows that the application needs the higher priority satellites to receive data earlier. The comm. module computes the link budget for a known set of specifications and protocols, and uses the resultant data rate to simulate DTN and compute bundle drop rates and latency to deliver any known bundle between any given pair of satellites. The optimization module ingests the outputs of the OM, ACS and Comm modules to compute the schedule of when each satellite should capture any POI, re-compute the future value of observing the said POI or region, and broadcast its decisions and observation-based insight. The schedule dictates the number of observations a satellite is expected to make over any region, which dictates the number, size and timing of data bundles generated for broadcast, therefore, we include a feedback loop between the optimizer and the comm. module. Slew characteristics depend on the previous GPs the satellite was observing and intended next, thus a feedback loop between the ACS and optimizer. If the constellation specifications, e.g. number of satellites, their arrangement, instruments, FOR, are expected to change over operational lifetime, a feedback between the OM and the optimizer may also be added.

Telemetry exchanged between the satellites is expected to contain updated information of constellation satellite states (S), seen GPs time series (Ω) or new GPs of interest, and their re-computed value individually or as a region (Δ). The mechanism of re-computing value at a POI (input on the left of Figure 1), i.e. onboard processing algorithms and predictive technology that will be applied to collected observations to predict the next observations and their associated value, are not in this paper's scope. We assume that processing will take a fixed time interval, and use a defined OSSE to determine the state of the next observations close to the last one. Also, we present a command and control scheduler, and assume that other operations, e.g. downlink, calibration, orbit maintenance, etc., will be scheduled separately and independently.

Nature Run for Observing System Simulations

We apply our proposed framework to episodic precipitation and resultant urban floods, to demonstrate its utility and scalability. Figure 3shows the frequency of floods in 1985 – 2010, as defined and tracked by the Dartmouth Flood Observatory (Brakenridge 2012). The redder the area, the more floods it has experienced. The points in red represent 42 large cities that are within these floodprone areas, with a 100 km radius buffer around them. We assume a 6 hour planning horizon, during which 5 of the 42

cities (Dhaka, Sydney, Dallas, London, Rio de Janeiro) flood to varying degrees as modeled by an OSSE nature run. For example, London tends to get slower, longer rains that might cause the Thames to flood, Dallas is more concerned with short, intense thunderstorms causing flooding on smaller creeks. The OSSE developed for this paper uses a square area of 80km x 80km, and is currently agnostic to flood-type disparities between cities.

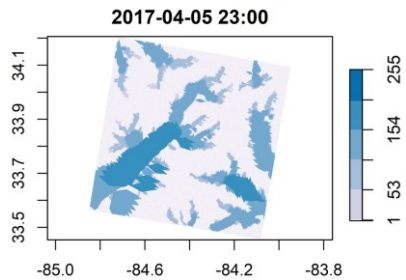

Figure 3 – Example of the spatial distribution of value (8-bit scale) of an ~80 km square region around Atlanta, GA (X/Y axis in degrees), for a single snapshot in time. The diamond-shaped watersheds, along whose stream-flow rate determines the value of the space-time observations, can be seen at some edges.

To describe the developed OSSE, we modeled riverine flooding in the Atlanta metropolitan area for a single storm event from April 5-6, 2017, using the WRF-Hydro hydrologic model version 5 (Gochis et al. 2018). WRF-Hydro links a land surface model (Niu et al. 2011) with terrain and channel flow routing modules to produce a continuous water balance, including streamflow. We supplied WRF-Hydro with meteorological forcing data from the North American Land Data Assimilation System (NLDAS) (hourly time step, 14x14 km spatial resolution). The model was run with a grid resolution of 900 m for the land surface model and 30m for the terrain and channel routing. The WRF-Hydro model was calibrated by adjusting spatial soil parameters and a soil infiltration tuning parameter until modeled streamflow matched measured flow at nine U.S. Geologic Survey gages in the Atlanta metro area. The modeled channel flow rates were then normalized by the 2-year recurrence interval flow rate (Q2), as estimated from the USGS regional regression equations for urban streams in that region (Feaster, Gotvald, and Weaver 2014). Q2 is the flow that has a 50% chance of happening in any year, and is an estimate of what constitutes a "flood" at any location. Finally, these normalized flood rates were then transformed on a log-scale to integer values between 1 and 256.

We estimated the watershed land area draining to each channel point and the [1,256] value of that point was assigned to the entire watershed area. Areas with high value correspond to watersheds with active flooding. In these watersheds, it is important to obtain satellite-derived estimates of precipitation to determine if this flooding will worsen (with more rainfall) or abate. This process provides an expected value of observing every point in a region of interest at 15 minute resolutions, to be used by the satellite scheduling optimizer. One time snapshot is shown in Figure 3. While this value function enforces preference in space-time where satellite-measured precipitation would be most valuable for assessing flood risk, *as satellites observe some GPs in a developing storm and consequent flood, the future value of the GPs in the region change*, because:

1. Observing a certain GP reduces the value of observing it and its immediate neighbors in the future, since computational models prefer a spread in space and time
2. Onboard processing of collected measurements (in a simplified version of the above process) allows satellites to update the value function

Thus, a high fidelity scheduler in a rapidly changing environment would need an implemented observing schedule to be taken into account, process data, update future value, re-compute its schedule accordingly and pass along insights and intent to the other satellites.

Orbital Mechanics and Attitude Control

We use homogeneous Walker Star-type constellation as our case study: 24 satellites arranged in 3 planes of 8 each. All satellites are simulated at a 710 km altitude, 98.5 deg inclination, circular orbits starting at epoch 1 January 2000. Instruments potentially used for precipitation and soil moisture sensing are narrow field of view (FOV) radars, which justify the need to continuously re-orient the <10 km footprint to cover a large flooding area. Examples are the Ka-band radar on the Global Precipitation Mission and a similar, smaller instrument called RainCube (Peral et al. 2015) for precipitation; or L-band and C-band radars on SMAP, P-band radar on Sentinal-1 and a Cubesat version of a P-band radar (Vega Cartagena et al. 2018) for soil moisture. This paper presents results for an 8km footprint instrument, however scalable to sub-km, if needed. The field of regard (FOR) which limits the maximum off-nadir angle of the payload/instrument is set to 55 deg, because it corresponds to 5x distortion of the nadir ground resolution, which is the OSSE's limit to allow combining observations in a given region. If the OSSE is sensitive to resolution, it can be incorporated into the value formulation, as demonstrated in a previous version of the framework.

The main revisions to the OM and ACS module since the previous version is that we now compute slewing time as a function of a pair of $sat_{t,s}$-gp_i , each representing a vector from satellite s at time t to ground point i. The dynamic programming (DP) algorithm in the optimizer now uses observable GPs as potential states, instead of discrete pointing options, as will be described in the subsequent

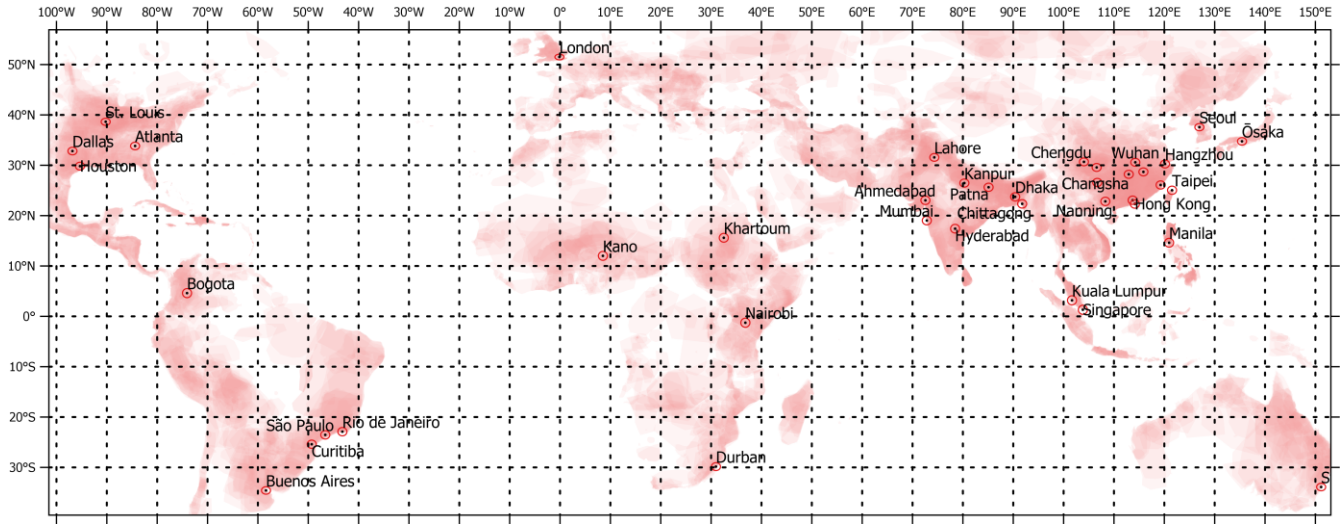

Figure 4 – Heat map of global floods between 1985 and 2011. Forty two cities have been identified from the most flooded urban regions.

sections. Since the DP optimizer iteratively calls for pairs of $sat_{t,s}$-gp_i to be evaluated, the ACS outputs cannot be pulled from a database dependent on satellite and fixed GP coordinates as before, and are computed real-time.

Onboard processing constraints limit our use of the full physics-based simulator (developed on MATLAB Simulink), therefore we developed weighted least squares on a third order polynomial whose coefficients are a function of the satellite mass, ACS and other specifications that served as knobs in the original model. The polynomial provides a very efficient implementation of the ACS-DP feedback loop in Figure 2, by allowing fast computation of slew time as a function of α, the angle between any pair of vectors $sat_{t,s}$-gp_i. This process is adaptable to non-planar angle dependencies as well, in case a full body re-orientation of the satellite is necessary, not just re-pointing an instrument.

The OM module has been developed in house, leveraging NASA GSFC's open-source General Mission Analysis Tool (Hughes 2007), and no longer depends on commercial products like AGI's STK. The OM also generates data bundle priority, to be ingested by the comm. module as follows: The data-bundle is tagged with the corresponding region or point of observation (where the data in the bundle is generated). Priority is given to the next satellite which can be a first to be able to access the same region or point, after the bundle generation, so that when it reaches the said region, it is up-to-date about its insights, as gathered by the last observing satellite. For example, if say Sat11 generates data over region "Dallas", and Sat12 is the next satellite on scene, Sat12 is given highest priority for the data-bundle to be delivered, and so on. When some satellites do not ever visit the region, they are removed from the priority queue.

Delay/Disruption Tolerant Networking

Delay/Disruption Tolerant Networking (DTN) is a new set of communication protocols developed with the intent of extending the Internet to users that experience long delays and/or unexpected disruptions in their link service. At its core, DTN defines an end-to-end overlay layer, termed "bundle layer", that sits on top of the transport layer (i.e., TCP or UDP in the Internet, CCSDS link layer standards in space) and efficiently bridges networks that may experience different types of operational environments. To achieve that, it encapsulates underlying data units (e.g., TCP/UDP datagrams) in bundles that are then transmitted from bundle agent to bundle agent, who store them persistently for safekeeping until the next network hop can be provisioned. This hop-to-hop philosophy is at the core of DTN and differentiates it from the Internet, where transactions typically occur by establishing end-to-end sessions (between the data originator and data sink).

At present, DTN is comprised of a large suite of specifications that encompass all aspects of network engineering, including its core protocols (e.g., the Bundle Protocol (Scott and Burleigh 2007), the Licklider Transmission Protocol (Farrell, Ramadas, and Burleigh 2008), or the Schedule Aware Bundle Routing (Standard and Book 2018)), adapters for bridging different types of underlying networks, network security protocols and network management functionality (Asynchronous Management Protocol). For the purposes of this paper, however, only the parts of the Bundle Protocol and Schedule Aware Bundle Routing Protocol were implemented. Together, they provide a medium to high fidelity estimate of how bundles would move in a DTN consisting of near-Earth orbiting spacecraft and allow us to quantify network figures of merit such as bundle loss or average bundle latency. Our DTN model is im-

plemented in Python using Simpy (Matloff 2008), a discrete-event engine built upon the concept of coroutines (or asynchronouos functions in the latest Python versions).

Dynamic Programming based Scheduler

Our proposed optimization algorithm uses dynamic program to step through the planning horizon, GPs in every region, and satellites in the constellation (mostly single, if running onboard). At any time point, it computes the best path of GP-observing options until that time such that the slewing times do not violate the sat-GP access constraints. The algorithm is comprehensively described in (Nag, Li, and Merrick 2017) with the key revision being – the state space that the path has to graph through is now time and GP, instead of time and satellite pointing direction (which mapped to observable GPs). By removing the necessity of discretizing the pointing space within FOR, the constraint of a minimum FOV is no longer needed and the algorithm is applicable to <4 deg FOV (e.g. radars). To ensure that memoization per node does not become astronomical, only the non-dominated paths (those with maximum cumulative value over space and time) are preserved at every step.

Summarized algorithm:

1: Inputs – sat, absval([gp_x,t_y]) *//OSSE output matrix* *[gp_x,t_y] $\forall x \in [1,numGP]$, $y \in [1,horiz_tSteps]$*
2: Output – $path_{sat}$[gp_i,t_i] *//vector[gp_i,t_i] $\forall i \in [1,pathLength]$, i.e. schedule of sat observing gp_i at t_i*
3: **For** tNow in planning_horizon **do**
4: **For** c in Constellation-{sat} **do**
5: $path_c$[gp_i,$t_i \leq tSrc_c$] ←commBundles(c,$tSrc_c$,sat,tNow)
6: **End For**
7: **For** s in satsWoverlappingFOR(sat) **do**
8: **For** gpNow in GroundPntsInFOR(sat, tNow) **do**
9: GroundPntsInBOX=[tNow-max(slewTime): tNow-min(slewTime),1:*numGP*]
10: **For** [gpBef,tBef] in GroundPntsInBOX **do**
11: v=computeValue($path_{sat}$[gpBef,tBef]+ [gpNow,tNow],absval,$path_{c \in Const}$[gp_i,$t_i \leq tSrc_c$])
12: **End For**
13: **For** vn in reverse_sort(v)
14: tslew=computeManueverTime([gp_1,t_1],[gp_2,t_2])
15: **If** tslew≤[t_2-t_1] **then**
16: $path_{sat}$[gpNow,tNow]← $path_{sat}$[gpBef,tBef]+[gpNow,tNow]
17: break // *forLoop for vn*
18: **End If**
19: **End For**
20: **End For**
21: **End for**
22: **End for**

Per the OSSE's time resolution, the cumulative value of observing any GP within 15min is constant, i.e. if it has been seen once, subsequent observations within 15min are of zero value. Since value re-computation based on collected data is not physically simulated, we estimate it from OSSE output (*absval*) in the following ways: The value of observing any GP after 15min is considered a fraction (= number of times it has been seen) of its OSSE-provided value with some random noise added. This re-computation ensures that a diversity of GPs is observed over time. Similarly, the value of observing a GP can be inversely proportionate to its distance to already observed GPs, to maximize the spatial spread of information collected and characterize the region better. The time-stepping nature of the algorithm causes it to be agnostic to the future value of any GP, therefore it can ingest changing values as they come along and compute schedules accordingly. We applied a standard normal distribution with a 2%-8% (uniformly random) standard deviation to the OSSE-provided values, to generate slightly different values to be used by each satellite's optimizer. This accounts for differences in onboard processing of their respective collections.

The advantages of this algorithm are:

1. Runtime is linearly proportionate to the number of time steps in planning horizon n(T). The scheduler needs to be rerun only if value of the GPs (expected to be accessed) change. If access is expected soon and a schedule is needed fast, n(T) can be shortened to span only the FOR access over the region.
2. Since the optimization proceeds monotonically through the planning horizon, it can be stopped at any given point in the runtime, and the resultant schedule is complete until the corresponding point in the horizon. Schedules can thus be executed as future schedules are being computed by the onboard processor.
3. A complex value function with non-linear dependencies (e.g. on viewing geometry or solar time) or multi-system interactions (e.g. Simulink or proprietary software calls) are easy to incorporate.
4. Algorithmic complexity per satellite per region to be scheduled is $O(n(S)\times n(T)\times n(GP)^2)$, where n(GP) is the number of ground points in the region and n(S) is the number of satellites *that can access the same GPs at the same time*. Since GPs are typically designed to Nyquist sample the footprint (down-sampled if the OSSE simulates a higher resolution), runtimes are instrument dependent.
5. If satellite FORs are non-overlapping, runtime or space complexity does not depend on the size of the constellation. Unless the constellation has tightly clustered satellites n(S) is likely only a couple.

Integer programming (IP) was able to verify that optimality of the above algorithm for non-overlapping regions was within 10%, and find up to 5% more optimal solutions, however the DP schedules were found at nearly four orders of magnitude faster than IP (Nag, Li, and Merrick 2017).

Results

The presented results are based on a simulation of 24 (20 kg cubic) satellites in a 3-plane Walker constellation observing floods in 5 global regions, as described. The selected specs allow for full body re-orientation of the satellite, however, our proposed algorithm can support gimbaled payload scheduling for heavier satellites by simply replacing the ACS model. The ACS model characterized with the satellite specs from (Nag, Li, and Merrick 2017) provides the following polynomial, where t is time for maneuver, and α is angle to span. The standard deviation is around 0.2116, so add 0.4232 to get ~95% percentile.

$$t = 6.1974 \times 10^{-6} \times \alpha^3 + 1.3904 \times 10^{-3} \times \alpha^2 + 1.4165 \times 10^{-1} \times \alpha + 4.6231$$

The 3-plane constellation in our case study provides a median 5.2 mins and a maximum 6.2 mins of access time (within FOR) to the regions of interest. While the gap between satellite accesses to a region, when there is an orbital plane overhead is ~10 mins, the 60 deg separation between the planes causes the overall median gap to be 56 mins and worst case ~4.5 hrs. Two orbital planes, with any number of satellites, would not be able to appropriately respond to a 6-hr flood phenomenon at all regions of interest even with agile pointing, crosslinks onboard autonomy. For the chosen altitude (Landsat, A-Train), at least 8 satellites per plane ensures consistent in-plane LOS, while cross-plane LOS is restricted to polar regions only. We will describe the effect of lower satellite numbers (e.g. 3 planes of 5 satellites each) and the performance effects of the lack of in-plane LOS in a future publication. We highlight the effect of the comm. module and optimizer on each other.

Performance of Inter-Satellite Networking

To estimate the performance of the DTN protocol stack, we first evaluated the supportable data rate in the inter-satellite links between spacecraft in the constellation. We make the following assumptions: All spacecraft transmit at S-band within the 6MHz typically available to class A missions; the link distance is set to 6000km for both intra and inter-plane links (from the OM module; we use the worst case for the inter-plane links since their distances are variable); all spacecraft are equipped with an SSPA that can deliver up to 5W of RF power, as well as a dipole placed parallel to the nadir/zenith direction. This design ensures minimal complexity since the SSPA can be directly connected to the antenna without needing splitters. Since the orientation of the spacecraft at any point in time is highly variable, we close the link budget assuming that both the transmitting and receiving antennas operate at the edge of the -3dB beamwidth. We consider that no atmospheric effects impair the links, and we select a ½ LDPC coding scheme together with a BPSK modulation, SRRC pulse shaping and NRZ baseband encoding. Using these inputs, we pessimistically estimate the link performance at 1kbps. Since multiple spacecraft can be in view of each other at any point in time (especially over the poles), and they all carry omnidirectional antennas, there is potential for interference problems that might occur at both the physical and the routing layer. For the former, we assume that signal interference is mitigated using some form of multiple access scheme (most likely Frequency or Code Division Multiplexing), but currently we do not model its performance. Therefore, the 1kbps data rate reported earlier must be interpreted as the data rate presented by the multiple access scheme to the upper layers of the protocol stack. Interference can also affect DTN's routing layer, discussed below.

To route data through the time-varying topology of the 24 satellite constellation, we simulate the system assuming that each of them is a DTN-enabled node with a simplified version of the Bundle Protocol and the Schedule Aware Bundle Routing Protocol. We assume that all communications are error-less at 1kbps, and that bundles have a size of 2000 bits - 1645 bits of payload data (provided by the OSSE) plus a 20% of overhead due to the protocol stack.

The DTN simulation uses the following inputs: The OM-provided contact plan is the basis for all routing decisions, and specifies contact opportunities between all the satellites in the network as a six element tuple: Start time, end time, origin, destination, average data rate, range in light seconds. Second, the traffic generated in the constellation, provided by the optimizer as a function of average collections, indicating when bundles are created, who sources them, who they are destined for, and the OM-provided relative priority flag with 14 levels. These priority levels are also used to set the Time-To-Live TTL property of all bundles according to the following rules: Priority 1 has a 15min TTL, priorities 2 and 3 have a 30min TTL, and priorities 4 to 15 have a 50min TTL. These rules let the network automatically discard stale information and minimize traffic congestion.

End-to-end latency experienced by 8341 bundles generated and sent over a 6 hour simulation (<1min DTN runtime) is shown in Figure 5. This latency is computed on a bundle-per-bundle basis, and measures the absolute time difference between the instant a bundle is delivered to the destination's endpoint (akin to TCP port), and the time it was originally created. Assuming a perfect multiple access scheme, any spacecraft might receive a copy of a bundle that was not originally intended for it, causing the problem of packet duplication in the system due to physical interference. If not dealt with, these extra copies would be rerouted and create exponential replication problem that

would overwhelm the entire system. To mitigate this, we take advantage of the extension blocks defined in the Bundle Protocol (Scott and Burleigh 2007). Particularly, every time router decides the next hop for a bundle, it appends an extension block with the identifier of the intended next hop. If another spacecraft receives a copy inadvertently, then the router simply discards it.

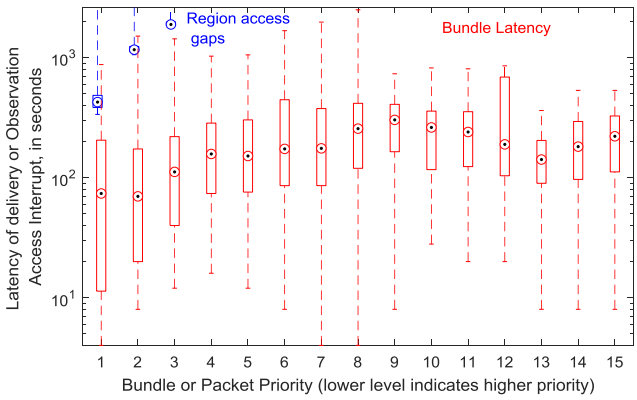

Figure 5 – Latency of data bundle delivery over all satellite pairs compared to the gaps between satellite FOR access to any region. If longest latency is less than shortest gap, for pairs with the same priority, each satellite can be considered fully updated with information from all others and perfect consensus is possible, in spite of distributed decisions made on a disjoint graph. Each box represents the 25%-75% quartiles, the circle is median, the whiskers show the max/min latency experienced or gap allowed.

Results indicate that latency is indeed affected by the bundle prioritization, however the effect is not monotonic because prioritization only happens at the bundle layer (e.g., radios have queues of frames, but they do not know about priorities in upper layers). Bundles with priorities 1-6 typically experience latencies of ~10s, with some outliers upto 15 minutes. This is quick enough for most of any satellite's knowledge to be transferred to the next two approaching any region (Figure 5). Also, no high priority bundles were dropped due to TTL expiration. Bundle with higher lower priorities experience increasingly large latencies of up to 2 minutes approximately on average, however never more than 15 minutes. The time to reach a region for by satellites with priority>3 is long enough for all bundles to be delivered for perfect consensus; in fact priority>4 is out of Figure 5's Y-axis range.

Performance of the Imaging Scheduler

The DP-based optimizer ingests outputs from all modules to find the best observational path that maximizes cumulative value till any given time, per satellite. We compare results from the use case in running the proposed algorithmic framework in 2 scenarios. One, the scheduler runs on onboard and uses collected information from other satellites as they come through the DTN. Two, the scheduler runs on the ground and uses collected information from other satellites as they downlink. The ground stations are placed at both poles to emulate the best possible scenario of collection-based re-computation twice an orbit.

Table 1—Comparison of optimizers run Onboard vs. Ground. A constellation with no agility sees 8.4% of the GPs, in either case)

	Scenario#1 (Distributed)	**Scenario#2 (Centralized)**
Cumulative Value (6h)	26120	21820
% of all GP observed	95.2%	99.2%

The distributed run uses re-computed value of any GP, based on all bundles about that GP (containing collection time and content) that have arrived. It risks knowing everything or nothing about any collection, based on DTN's performance. Our implementation shows that the constellation predicts GP value at an average of 4% different from actual value, due to bundles about GPs arriving later than the satellite already observes them. This happens only for an outlier of one or two hop connections (Figure 5), thus 95.2% of the GPs in all 5 regions are observed. Longer the DTN latency, larger the perceived to actual value difference per satellite, lower the cumulative collected value.

The centralized run has no risk of low-value or redundant observations, e.g. due a satellite not knowing a GP or its neighbors were seen 20 mins ago. Space-time intent info is inherently synced (without DTN dependency), allowing for >99% of GPs seen. However, value functions are based on data collected approximately an orbit earlier, due to collection and up/downlink latency between any satellite pair. Our implementation shows that the constellation predicts GP value at an average of 70% different from actual value, due to the difference in satellite collections. While this difference is a function of synthetic OSSE variation, future work includes physics based models of higher fidelity. Nonetheless, it shows that in fast changing environments, the performance benefit of a responsive constellation is obvious. The DTN-enabled decentralized solution provides 16% more value over 6 hours than the centralized implementation of the same algorithm.

The time taken to run the algorithm per satellite was 1% of the planning horizon time, as evaluated on MATLAB installed in a Mac OS X v10.13.6 with a 2.6 GHz processor and 16 GB of 2400 MHz memory.

Acknowledgements

This work has been supported by the NASA New Investigator Program, Earth Science Technology Office, and the Interplanetary Network Directorate at the Jet Propulsion Laboratory.

References

Abramson, Mark R., Stephan Kolitz, Eric Robinson, and Dorri Poppe. 2013. "Earth Phenomena Observation System (EPOS) for Coordination of Asynchronous Sensor Webs." In *AIAA Infotech@ Aerospace (I@ A) Conference*, 4813. https://arc.aiaa.org/doi/abs/10.2514/6.2013-4813.

Arnold Jr, Charles P., and Clifford H. Dey. 1986. "Observing-Systems Simulation Experiments: Past, Present, and Future." *Bulletin of the American Meteorological Society* 67 (6): 687–695.

Barnsley, M. J., J. J. Settle, M. A. Cutter, D. R. Lobb, and F. Teston. 2004. "The PROBA/CHRIS Mission: A Low-Cost Smallsat for Hyperspectral Multiangle Observations of the Earth Surface and Atmosphere." *Geoscience and Remote Sensing, IEEE Transactions on* 42 (7): 1512–1520.

Bianchessi, Nicola, and Giovanni Righini. 2008. "Planning and Scheduling Algorithms for the COSMO-SkyMed Constellation." *Aerospace Science and Technology* 12 (7): 535–544.

Brakenridge, G. R. 2012. *Global Active Archive of Large Flood Events, Dartmouth Flood Observatory, University of Colorado*.

Cahoy, Kerri, and Andrew K. Kennedy. 2017. "Initial Results from ACCESS: An Autonomous CubeSat Constellation Scheduling System for Earth Observation." In . Logan, Utah. http://digitalcommons.usu.edu/smallsat/2017/all2017/98/.

Cerf, Vinton, Scott Burleigh, Adrian Hooke, Leigh Torgerson, Robert Durst, Keith Scott, Kevin Fall, and Howard Weiss. 2007. "Delay-Tolerant Networking Architecture."

Damiani, Sylvain, Gérard Verfaillie, and Marie-Claire Charmeau. 2005. "An Earth Watching Satellite Constellation: How to Manage a Team of Watching Agents with Limited Communications." In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, 455–462. ACM. http://dl.acm.org/citation.cfm?id=1082543.

Eickhoff, Jens. 2011. *Onboard Computers, Onboard Software and Satellite Operations: An Introduction*. Springer Science & Business Media. https://books.google.com/books?hl=en&lr=&id=JUYo2E9UB18C&oi=fnd&pg=PP2&dq=jens+eickhoff+onboard&ots=pzlK4F7Qt1&sig=wlngMSJIL3uI_tEeCjc3ndNHzmQ.

Farrell, Stephen, Manikantan Ramadas, and Scott Burleigh. 2008. "Licklider Transmission Protocol-Security Extensions."

Feaster, Toby D., Anthony J. Gotvald, and J. Curtis Weaver. 2014. "Methods for Estimating the Magnitude and Frequency of Floods for Urban and Small, Rural Streams in Georgia, South Carolina, and North Carolina, 2011." *US Geological Survey, SIR* 5030.

Feldman, Daniel R., Chris A. Algieri, Jonathan R. Ong, and William D. Collins. 2011. "CLARREO Shortwave Observing System Simulation Experiments of the Twenty-First Century: Simulator Design and Implementation." *Journal of Geophysical Research: Atmospheres (1984–2012)* 116 (D10). http://onlinelibrary.wiley.com/doi/10.1029/2010JD015350/full.

Frank, Jeremy, Minh Do, and Tony T. Tran. 2016. "Scheduling Ocean Color Observations for a GEO-Stationary Satellite." In *Proceedings of the Twenty-Sixth International Conference on International Conference on Automated Planning and Scheduling*, 376–384. AAAI Press. http://dl.acm.org/citation.cfm?id=3038642.

Globus, Al, James Crawford, Jason Lohn, and Robert Morris. 2002. "Scheduling Earth Observing Fleets Using Evolutionary Algorithms: Problem Description and Approach." https://ntrs.nasa.gov/search.jsp?R=20020091594.

Gochis, D.J., Michael Barlage, A. Dugger, K. Fitzgerald, L. Karston, M. McAllister, J. McCreight, et al. 2018. "The WRF-Hydro Modeling System Technical Description, (Version 5.0)." NCAR Technical Note DOI:10.5065/D6J38RBJ.

He, Lei, Xiao-Lu Liu, Ying-Wu Chen, Li-Ning Xing, and Ke Liu. 2019. "Hierarchical Scheduling for Real-Time Agile Satellite Task Scheduling in a Dynamic Environment." *Advances in Space Research* 63 (2): 897–912. https://doi.org/10.1016/j.asr.2018.10.007.

Hughes, Steven P. 2007. "General Mission Analysis Tool (GMAT)." http://ntrs.nasa.gov/search.jsp?R=20080045879.

Jian, Li, and Wang Cheng. 2008. "Resource Planning and Scheduling of Payload for Satellite with Genetic Particles Swarm Optimization." In *Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on*, 199–203. IEEE. http://ieeexplore.ieee.org/abstract/document/4630799/.

Lemaître, Michel, Gérard Verfaillie, Frank Jouhaud, Jean-Michel Lachiver, and Nicolas Bataille. 2002. "Selecting and Scheduling Observations of Agile Satellites." *Aerospace Science and Technology* 6 (5): 367–381.

Lin, Wei-Cheng, Da-Yin Liao, Chung-Yang Liu, and Yong-Yao Lee. 2005. "Daily Imaging Scheduling of an Earth Observation Satellite." *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 35 (2): 213–223.

Martin, William. 2002. "Satellite Image Collection Optimization." *Optical Engineering* 41 (9): 2083–2087.

Matloff, Norm. 2008. "Introduction to Discrete-Event Simulation and the Simpy Language." *Davis, CA. Dept of Computer Science. University of California at Davis. Retrieved on August* 2 (2009): 1–33.

Muraoka, Hiroyasu, Ronald H. Cohen, Taijiro Ohno, and Naoyuki Doi. 1998. "Aster Observation Scheduling Algorithm." In *Proceedings of the International Symposium Space Mission Operations and Ground Data Systems*. Citeseer. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.501.9647&rep=rep1&type=pdf.

Nag, Sreeja, Charles K. Gatebe, and Olivier de Weck. 2015. "Observing System Simulations for Small Satellite Formations Estimating Bidirectional Reflectance." *International Journal of Applied Earth Observation and Geoinformation* 43: 102–18. https://doi.org/10.1016/j.jag.2015.04.022.

Nag, Sreeja, Charles Gatebe, D. W. Miller, and O. L. De Weck. 2016. "Effect of Satellite Formation Architectures and Imaging Modes on Global Albedo Estimation." *Acta Astronautica* 126 (April): 77–97. https://doi.org/10.1016/j.actaastro.2016.04.004.

Nag, Sreeja, Alan Li, and James Merrick. 2017. "Scheduling Algorithms for Rapid Imaging Using Agile Cubesat Constellations." *COSPAR Advances in Space Research - Astrodynamics*, July.

Niu, Guo-Yue, Zong-Liang Yang, Kenneth E. Mitchell, Fei Chen, Michael B. Ek, Michael Barlage, Anil Kumar, Kevin Manning, Dev Niyogi, and Enrique Rosero. 2011. "The Community Noah Land Surface Model with Multiparameterization Options (Noah-MP): 1. Model Description and Evaluation with Local-Scale Measurements." *Journal of Geophysical Research: Atmospheres* 116 (D12).

Peral, Eva, Simone Tanelli, Ziad Haddad, Ousmane Sy, Graeme Stephens, and Eastwood Im. 2015. "Raincube: A Proposed Constellation of Precipitation Profiling Radars in CubeSat." In *Geoscience and Remote Sensing Symposium (IGARSS), 2015 IEEE International*, 1261–1264. IEEE. http://ieeexplore.ieee.org/abstract/document/7326003/.

Robinson, Eric, Hamsa Balakrishnan, Mark Abramson, and Stephan Kolitz. 2017. "Optimized Stochastic Coordinated Planning of Asynchronous Air and Space Assets." *Journal of Aerospace Information Systems*. https://arc.aiaa.org/doi/full/10.2514/1.I010415.

Sandau, R., H. P. Roeser, and A. Valenzuela. 2010. *Small Satellite Missions for Earth Observation: New Developments and Trends*. Springer Verlag. http://books.google.com/books?hl=en&lr=&id=D6LaoU-VsxQC&oi=fnd&pg=PR6&dq=sandau+small+satellite+missions+for+earth+observation&ots=VjuMKOSfDa&sig=reMkSbdiHAfAuUWUpH8X5D8FupQ.

Scott, K., and S. Burleigh. 2007. "RFC 5050: Bundle Protocol Specification." *IRTF DTN Research Group*.

Shao, Elly, Amos Byon, Chris Davies, Evan Davis, Russell Knight, Garett Lewellen, Michael Trowbridge, and Steve Chien. 2018. "Area Coverage Planning with 3-Axis Steerable, 2D Framing Sensors." In *Scheduling and Planning Applications Workshop , International Conference on Automated Planning and Scheduling*. Delft, The Netherlands.

Sherwood, Rob, Anita Govindjee, David Yan, Gregg Rabideau, Steve Chien, and Alex Fukunaga. 1998. "Using Aspen to Automate Eo-1 Activity Planning." In *Aerospace Conference, 1998 IEEE*, 3:145–152. IEEE. http://ieeexplore.ieee.org/abstract/document/685789/.

Standard, Proposed Draft Recommended, and Proposed Red Book. 2018. "Schedule-Aware Bundle Routing."

Vega Cartagena, Manuel A., Jeffrey R. Piepmeier, James Garrison, Joseph J. Knuble, Cornelis F. Du Toit, and Matthew A. Fritts. 2018. "Signals of Opportunity-Airborne Demonstrator (SoOP-AD): Instrument Overview, Performance during First Flights and Future Instrument Concept [STUB]."

Xhafa, Fatos, Junzi Sun, Admir Barolli, Alexander Biberaj, and Leonard Barolli. 2012. "Genetic Algorithms for Satellite Scheduling Problems." *Mobile Information Systems* 8 (4): 351–377.