Designing a Disruption Tolerant Network for Reactive Spacecraft Constellations

Sreeja Nag[1]
NASA Ames Research Center, BAER Institute, Moffet Field, CA 94035, U.S.A.

Marc Sanchez Net[2]
NASA Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, CA 91109

Alan S. Li[3] and Vinay Ravindra[4]
NASA Ames Research Center, BAER Institute, Moffet Field, CA 94035, U.S.A.

Small spacecraft can now support operational agility due to cognitive payloads, tunable duty cycles, and precise attitude control systems that can re-orient the spacecraft and capture images within short notice. When combined with onboard processing and autonomous scheduling 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, scheduling optimization and some preliminary inter-spacecraft communications to plan the time-varying, full-body orientation of agile, small spacecraft in a constellation. The proposed schedule optimization can run autonomously onboard the spacecraft without ground control, or at the ground station with resultant schedules uplinked to the spacecraft for execution. This paper describes the design of the communication module, which is based on Delay/Disruption Tolerant Networking (DTN) for onboard data management and routing among the satellites. The physical layer has been modeled with small satellite radios, with appropriate amplifiers, coding and modulation. The data link layer's interference concerns have been addressed using three Multiple Access schemes, to mitigate a data rate penalty within and across constellation planes. The combined framework been applied to representative constellations making targeted measurements of episodic precipitation events and subsequent urban floods. Results on a 24-satellite constellation observing 5 global regions show appropriately low latency in information exchange (average within 1/3rd available time for implicit consensus), enabling the onboard scheduler to observe ~7% more flood magnitude than a ground-based implementation. Both onboard and offline versions performed ~98% better than constellations without agility.

I. Acronyms

DTN	Delay/Disruption Tolerant Networking
LEO	Low Earth Orbit
MF-FDMA	Multi-Frequency Frequency Division Multiple Access
OM	Orbital Mechanics (module)
TTL	Time to Live (for DTN packets)

[1] Senior Research Scientist, BAER Institute/NASA Ames Research Center, and AIAA Senior Member.
[2] Telecommunications Engineer, Jet Propulsion Laboratory, and AIAA Member
[3] Research Scientist, BAER Institute/NASA Ames Research Center
[4] Research Scientist, BAER Institute/NASA Ames Research Center, and AIAA Member

II. Introduction

Small spacecraft and the proliferation of launch service providers has lowered the cost of access to space, allowing distributed space missions (DSMs) to complement unmanned aerial systems (UAS) and vast ground-based networks in building an agile Sensor Web [1] for Earth Observation. Large numbers of small spacecraft can maximize *operational agility* if provided with adequate onboard processing resources for automated inference and scheduling, and Inter-Satellite Links (ISLs) for exchanging information about collected data, telemetry, command and control[2]. Spacecraft may be operationally agile from a payload internal perspective, e.g., selection of a frequency channel or transmit power, or in a physical sense, e.g., re-orient in three degrees of freedom (DOF) to re-point their instruments within short notice (e.g. CHRIS on Proba [3]). Small spacecraft do have onboard processing to interpret collected science data and potentially improve their observing plans (e.g. IPEX Cubesat as a HyspIRI pathfinder [4]), and have demonstrated inter-sat communication links to transmit data or metadata (e.g. NODeS on a pair of Edison-like Cubesats [5]). Schedulers for Cubesat constellations such as the 200+ Dove spacecraft fleet operated by Planet Labs [6] or for the 3-DOF imaging for their agile Skybox spacecraft [7] have been published by industry players. Academic literature has shown successful scheduler simulations for step-and-stare approaches by matrix imagers [8], mixed integer programming optimization [9], adaptive large neighborhood searching [10], and real-time re-computation of science value to plan for constellation satellite re-orientation to observe transient or evolving EO phenomena[2]. If such schedulers can be adapted for onboard execution, the addition of *optimized* inter-sat networking to the constellations is guaranteed to improve their operational agility, especially in the absence of quick turnaround times by ground station contacts or humans in the loop. This allows spacecraft to respond to observations by making inferences and transmitting across the constellation, so that all spacecraft can respond appropriately in coordination. Applications for such responsiveness can range from time-critical tracking of airplanes[11], monitoring fast phenomena like the spread of tropical cyclones[12], near-simultaneous coordination for multi-angular remote sensing[13], [14], and planetary missions that do not have the privilege of regular Earth communication[15].

Power and bandwidth restrictions on small spacecraft has spurred literature on scheduling data download from/to a network; e.g., optimization of single Cubesat downlink to a network of ground stations (GS) and multiple payloads' downlink to existing stations [16] within available storage, energy and access time constraints. While some[17] use crosslinks to propagate planning information through the constellation, the tools are optimized for data downlink. Since they are agnostic to the data content (only size matters), payload type and concepts of operations, they are not particularly appropriate for custom Earth or planetary cience applications. Planners that negotiate task assignment but without realistic orbital constraints [18], [19], or simulate the space environment but not realistic inter-satellite communication [20], are far more common than those that consider both factors. Recent studies[21], [22] on scheduling duty cycles of a constellation of radars has shown science benefits of Inter-Satellite-Links (ISLs) for command and control, however the communication module is modeled at the physical layer only, which is insufficient as the data, applications and networks scale.

This paper describes the implementation of the delay/disruption tolerant network (DTN) paradigm [23], an emerging protocol suite for routing data in dynamic and intermittent operational environments such as small spacecraft EO constellations monitoring transient precipitation and urban floods. The implementation accounts for constraints from the physical and data link layers, and shows communication latencies that are acceptable for inter-spacecraft coordination. The DTN paradigm [24], [25] provides a store-and-forward mechanism that ensures reliable data delivery via a schedule-aware bundle routing mechanism that uses a 'contact plan' describing the current and future connectivity of the network to obtain a routing solution. While DTN has demonstrated benefit for planetary rover operations [15], it has never been applied to EO scheduling, for which it is appropriate because of the dynamic nature of Low Earth Orbit (LEO) constellations. For example, DTN can minimize replication and improve the delivery probability within available resources of power and bandwidth. In the event of unplanned data generation and/or unexpected link outages, a DTN node can exercise hop-by-hop routing schemes (e.g. Dijkstra algorithm) to deliver the data.

III. Concept of Operations

We demonstrate the implementation of a communication network to improve coordination in a satellite constellation, which can then generate more valuable measurements of fast changing precipitation and urban floods. Figure 1 shows the interaction of the communication module with the other modules, specifically orbital mechanics (OM) and Attitude Control Systems (ACS), within the framework, which then optimizes a schedule for any satellite (sat) in the constellation, to observe a known set of ground regions with rapidly evolving phenomena. The schedule optimizer may run onboard satellites, which generate data bundles after executing scheduled observations and

broadcast using the DTN. Bundles contain information about the ground points observed and meta-data from an onboard processing software and/or science simulator. Considering network delays in a temporally varying disjoint graph, fast-changing access to regions (sat ground velocity ~7 km/s) and fast-changing phenomena, satellites are not expected to iterate on acknowledgments to establish explicit consensus. Instead, we aim at implicit consensus; the more a satellite knows about a region before its observation opportunity, better its scheduler performance. The schedule optimizer may also 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 satellites. Since the ground stations are expected to be inter-connected on Earth and in sync with each other at all times, the optimization is centralized and the resultant schedule avoids potentially redundant observations due to lack of consensus among the satellites. It also reduces onboard processing requirements. However, since information relay occurs at only sat-ground contacts (function of orbits, ground network), the scheduler may use significantly *outdated* information compared to the distributed, onboard bundles. The transiency of the phenomenon being observed, robustness to latency in exchanging inferences determines effectiveness of the onboard, decentralized vs. ground, centralized implementation of the scheduler.

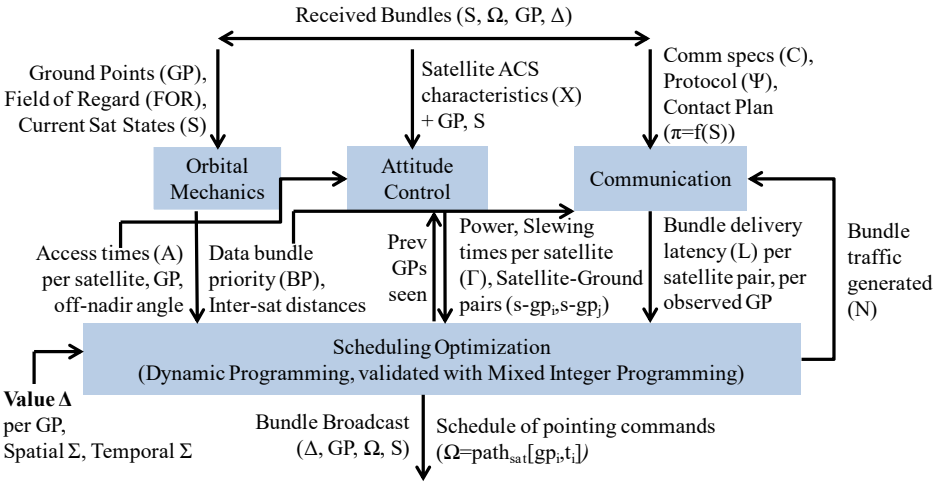

Figure 1: Major information flows between the modules in the proposed agile EO scheduler, expected to run onboard every satellite in a given constellation, applied to global urban flooding in this paper. This framework can exchange information (as identified at the top) between the satellites via peer-to-peer communication or via the ground (reverse bent pipe architecture).

The OM and ACS, modules in the Figure 1 framework are described comprehensively in [2],[26],[27]. To summarize, the OM module propagates orbits in a known constellation and computes possible coverage of known regions of interest (appropriately discretized into Grid Points (GPs)) within a Field of Regard (FOR is the angular area a sat *can* observe by re-orienting). It also provides line-of-sight (LOS) availability, inter-sat distances, and the priority of bundle delivery to the Communication (includes ISL) module between every pair of satellites. For example, if Sat1 generates data over Dallas, and Sat2 is the next to access Dallas, Sat2 is given highest priority. The ACS module computes the time required by any satellite to slew from one GP to another (including satellite orbital movement), resultant power, momentum and stabilization profiles. As this paper will go on to describe, the comm. module computes the link budget for known satellite specifications (e.g. radio power, antenna size) in the physical layer. The resultant data rate is then used as an input to simulate a DTN at the network layer, and compute latency to deliver any bundle between any given pair of satellites. The scheduler ingests the outputs of the OM, ACS and Comm modules and outputs the schedule $path_{sat}[gp_i,t_i]$, an array of tuples $[gp_i,t_i]$ which represents when (t_i) a *sat* should capture any gp_i. The executed schedule dictates the number of observations that a satellite makes, which then dictates the number, size and timing of bundles generated for broadcast. Therefore, the framework has a feedback loop between the scheduler and comm. Module, to capture these cross-dependencies. Slew characteristics depend on the previous GPs the satellite was observing and intended next, thus a feedback loop between ACS and scheduler.

IV. Communication System Design

While the proposed communication solution scales across constellation topologies and observation requirements[28], we use a baseline constellation in this paper to describe the communication system that can respond to sub-hour transient natural phenomena. The baseline is a 3-plane Walker constellation observing floods in 5 global regions over a 6 hour simulation duration. Each plane contains 8 equally spaced (20 kg cubic) satellites, i.e., 24 in total. The 3-plane constellation provides a median 5.2 mins and a maximum 6.2 mins of access time (within field of regard) to the regions of interest. The gap between satellite accesses to a region when there is an orbital plane overhead is ~10 mins, but 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 (at which the flagship missions Landsat and A-Train fly), at least 8 satellites per plane ensures consistent in-plane LOS, while cross-plane LOS is restricted to polar regions only. All satellites are simulated at a 710 km altitude, 98.5 deg inclination, circular orbits starting at epoch 1 January 2000.

We used a narrow field of view (NFOV) radars with an 8km footprint for the satellite payload, based on the RainCube [29] mission. Floods are not observed directly, and are predicted onboard as a function of observed precipitation. The science simulation is described in more detail in Ref[2], with some updates to the prediction and objective functions. Radars are generally NFOV, therefore need to continuously re-orient their FOV to cover a large flooding area. 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, beyond which it is difficult to combine observations in a given region. The five regions assumed to flood simultaneously were {Dhaka, Sydney, Dallas, London, Rio de Janeiro}, each modeled by discretized GPs within a square area of 80km x 80km. Results will show impact of expanding to forty regions globally.

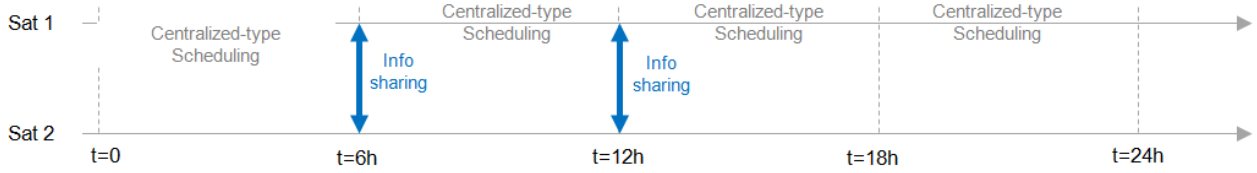

Figure 2: The baseline scheduler plans observations for all satellites in a centralized manner over a pre-defined horizon after information on executed schedules has been synchronized across the constellation (every 6 hours)

The baseline scheduler used in this paper uses greedy, dynamic-programming to select observations ($path_{sat}[gp_i,t_i]$) for every satellite. While the solutions are not as optimal as a mixed integer programming approach[26], they are suitable for real-time execution onboard[2]. The scheduler plans observations at a regular cadence (e.g. every 6 hours in Figure 2) using the information that it has received from the DTN network, on other satellite observations and inferences, since the last time it scheduled. The planning horizon is at least the duration of the cadence (e.g. 6 hours) but may be longer to allow for more optimality at the cost of responsiveness. The quality of the schedule is therefore impacted by how informed the said scheduler is about the other satellites' executed schedules and planned schedules (within immutable duration), which is impacted by the latency of bundle delivery since the bundle contains observations or inferences. This dependence is true not just for the baseline scheduler, but for any scheduler (Section II) that re-plans using new information.

The following sub-sections outline three layers of the communication module. The *physical layer* is constrained by small sat systems and achievable data rates using low gain antenna. High gain antennas that yield better link performance were also initially considered. However, the need to gimbal the antenna forces us to couple the problems of observation and link scheduling, rendering the problem complex and potentially restricting satellite observation periods. The *data link layer* is modeled because a multiple access scheme is required to avoid collisions in shared spectrum for large constellations. The *network layer* adopts automated routing of information within the constellation using DTN's Contact Graph Routing to deliver all bundles generated by the above case study simulation. Estimation of accurate end-to-end latency across the network is an important output of the DTN simulation since it constraints the scheduling algorithm. Utilization of external constellation assets such as the U.S. Tracking and Data Relay Satellite System (TDRS) may be considered to improve performance[30].

A. Physical Layer Constraints and Trades

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. Initial analysis of the baseline constellation showed the intra-plane distance to be nearly constant at ~5425km and the inter-plane distance to vary between 80 km (near pole crossings) and 6060 km, making data-rates very dependent on when and which satellite would like to communicate. We make the following assumptions: All spacecraft carry the same radio with single data rate capability. They 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; this could be set as a customized variable from the OM module, however we use the worst case for the inter-plane links since their distances are variable. All spacecraft are equipped with a Solid State Power Amplifier (SSPA) that can deliver up to 20W of RF power, as well as a dipole placed parallel to the nadir/zenith

	Value	Units	Id
SPECTRUM CONFIGURATION			
Mission category	A	-	1
Frequency band	S	-	2
Network support	NTIA/FCC	-	3
Support service	Any	-	4
Carrier frequency	*2.30E+03*	MHz	5
Carrier wavelength	*1.30E-01*	m	6
Max. usable bandwidth	*6.00*	MHz	7
LINK CONFIGURATION			
Receiver type	Custom	-	8
Modulation	BPSK	-	9
Code rate	1/2	-	10
Baseband encoding	NRZ	-	11
Pulse shaping	SRRC	-	12
Shaping expansion factor	*1.25*	-	13
Modulation index	90.00	deg	14
Link distance	6000.00	km	15
User data rate	1.00E+03	bps	16
Desired link margin	3.00	dB	17
TRANSMITTER PARAMETERS			
Tx power	13.01	dBW	18
Tx antenna gain	2.15	dBi	19
EIRP	*15.16*	dBW	20
Tx line/waveguide loss	0.70	dB	21
Tx pointing loss	2.15	dB	22
Radiated power	*12.31*	dBW	23
PATH PARAMETERS			
Space loss	*175.24*	dB	24
Atmospheric losses	0.10	dB	25
Polarization losses	0.10	dB	26
Total path losses	*175.44*	dB	27
RECEIVER PARAMETERS			
Rx pointing loss	2.15	dB	28
Rx antenna gain	*2.15*	dBi	29
Rx system noise temperature	*28.45*	dB-K	30
Rx G/T	*-26.30*	dB-K	31
POWER SUMMARY			
Boltzmann constant	*-228.60*	dBJ-K	32
Noise spectral density (No)	*-200.15*	dBW-Hz	33
Received C/No	*37.02*	dB-Hz	34
Carrier suppression loss	*0.00*	dB	35
Received data C/No	*37.02*	dB-Hz	36
Data rate	*30.00*	dB-bps	37
Received Eb/No	*7.02*	dB	38
CODING AND BANDWIDTH SUMMARY			
Symbol per modulated symbol	*1*	-	39
Coded symbol rate	*2.00E+03*	sps	40
Analog signal BW	*0.00*	MHz	41
Available BW	*6.00*	MHz	42
BW occupancy	0.04%	-	43
Assumed coding gap	*1.31*	dB	44
DATA RATE SUMMARY (User inputs)			
AWGN capacity threshold	*0.19*	dB	45
Required Eb/No	*1.50*	dB	46
Rx Implementation Loss	2.00	dB	47
MA Loss	0.00	dB	48
Eb/No margin	3.52	dB	49

Figure 3: Link budget to determine data rate for 20W RF transmission. The baseline mission uses 5W RF transmission with all other factors remaining the same

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 to account for radio implementation losses. We consider that no atmospheric effects impair the links, and we select a ½ LDPC (low-density parity-check) coding scheme together with a Binary Phase Shift Keying (BPSK) modulation, SRRC (square root raised cosine) pulse shaping and NRZ baseband encoding; such coding schemes are standard practice for near-Earth missions. Using these inputs, we pessimistically estimate the link performance at 1kbps – see Figure 3 for the detailed link budget for 20W transmission. 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 impact both the physical and the network layer, namely the routing mechanism. For the former, we assume that signal interference is mitigated using some form of multiple access scheme, as discussed in Section IV.C. 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 in Section IV.B.

B. Data Layer Interference Considerations

Since multiple spacecraft can be in view of each other at any point in time and they carry omnidirectional dipole antennas, there is potential for interference both within and across constellation planes. Therefore, in this work we considered three Multiple Access (MA) schemes: CDMA (Code Division Multiple Access), MF-TDMA (Multi-Frequency Time Division Multiple Access) and MF-FDMA (Multi-Frequency Frequency Division Multiple Access). A detailed explanation of the analysis for each of these MA systems is beyond the scope of this paper, however their relative pros and cons have been summarized in Table 1.

Table 1: Comparison between the considered three Multiple Access (MA) schemes

	CDMA	MF-FDMA	MF-TDMA
Advantages	Heritage at NASA; e.g. TDRS uses it for MA and is standardized in CCSDS Simple to implement from the radio and software standpoint.	Typical way of implementing MA in space provided there is enough bandwidth for the application; e.g., commercial communication cannot use it. Separation of data from different users is simply achieved by filtering.	Simple frequency plan with few bandguards. User separation is achieved at frame level through software processing. Each radio only requires three tracking loops, thus reducing complexity.
Disadvantages	System performance is limited by interference power from other users transmitted simultaneously. Can interfere with TDRS MA users Losses grow faster than exponentially.	Requires enough bandguards to ensure Doppler shift in inter-plane communication results in out-of-band transmission. Increased radio complexity due many tracking loops needed. Requires dedicated S-band allocation	Requires time synchronicity which might be difficult to achieve in space. Frequency-Time duality only valid if underlying link data rate can be scaled up to compensate for pause times (which is not true for power constrained links). Requires dedicated S-band allocation

CDMA mitigates interference by using a spread-spectrum signal, generated by performing an XOR operation between the data bit sequence and a unique chip sequence pre-assigned to each spacecraft. Therefore, the chip rate is limited by NTIA bandwidth allocations, and results in a rate of ~3Mcps for an S-band constellation with SRRC pulse shaping and 7.5MHz bandwidth available (based on TDRSS MA return service). The performance of the CDMA system depends on self-interference, i.e., the number of users transmitting at the same time. Figure 4 confirms that up to 1.5 dB of self-interference is possible since at most 8 satellites can be in view of one another at certain points in time (at the poles). This is too high a penalty given that the link rate (without taking it into account) has already been constrained to 1 kbps, therefore the CDMA scheme is not preferable.

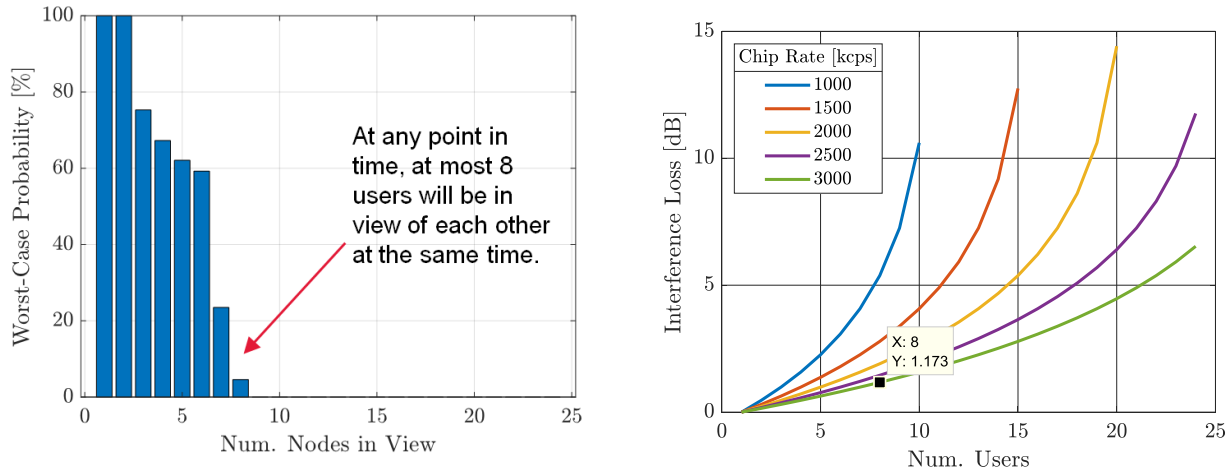

Figure 4: Histogram of satellites in view by any other in the constellation (left) and CDMA interference loss due to satellites in view for a given chip rate (right). The chosen operating point has been highlighted.

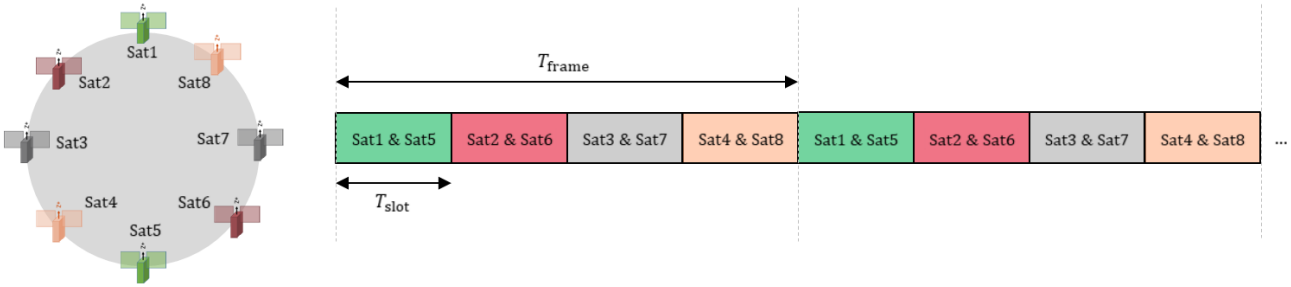

Figure 5: Time multiplexing of eight satellites per plane to prevent MF-TDMA interference results in 4 slots per frame and 4x reducion in data rate. All other combinations will interfere or be wasteful.

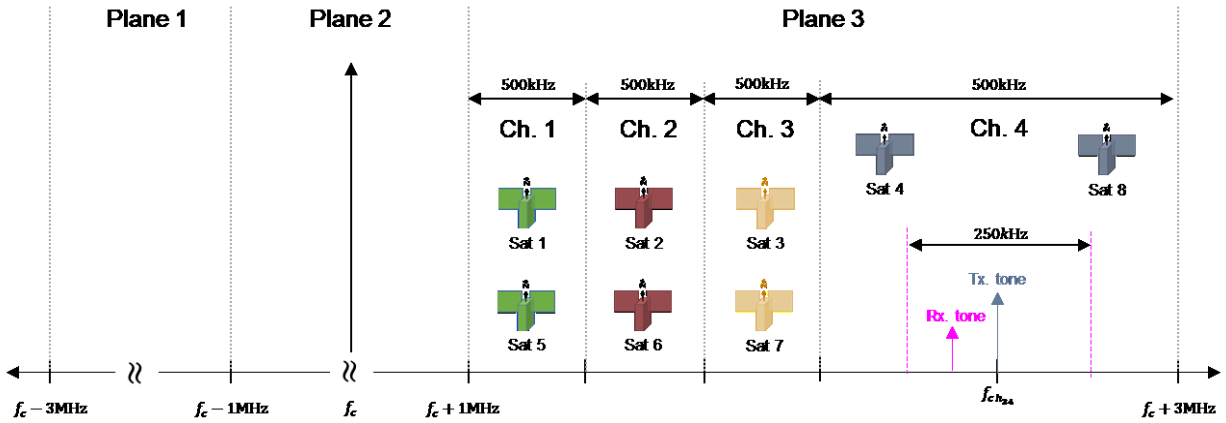

Figure 6: Frequency Plan for MF-FDMA

In MF-TDMA, spacecraft within a plane share spectrum by time-multiplexing their transmissions. Therefore, only one frequency per plane is needed and this reduces the total required bandwidth for the system to work. MF-TDMA was also considered unfeasible because at least four slots are needed to avoid intra-plane interference, owing to the infeasibility of two adjacent satellites in the same plane transmitting in the same time slot (separation across planes is achieved through frequency). Figure 5 shows the cartoon representation of 8 satellites in a plane and the division of a frame into four slots to avoid interference, thereby reducing the effective data rate per user satellite by the same factor.

Typically, one would compensate this loss by increasing the data rate by 4x, however this is not possible in an already constrained link budget. Therefore, a 6 dB of penalty is incurred and an effective rate of just 250 bps is left for data transmission, deeming MF-TDMA to be a less preferred scheme.

	Value	Units	Id
SPECTRUM CONFIGURATION			
Mission category	A	-	1
Frequency band	S	-	2
Network support	NTIA/FCC	-	3
Support service	Any	-	4
Carrier frequency	2.30E+03	MHz	5
Carrier wavelength	1.30E-01	m	6
Max. usable bandwidth	6.00	MHz	7
LINK CONFIGURATION			
Receiver type	Custom	-	8
Modulation	BPSK	-	9
Code rate	1/2	-	10
Baseband encoding	NRZ	-	11
Pulse shaping	SRRC	-	12
Shaping expansion factor	1.25	-	13
Modulation index	90.00	deg	14
Link distance	6000.00	km	15
User data rate	8.50E+02	bps	16
Desired link margin	3.00	dB	17
TRANSMITTER PARAMETERS			
Tx power	13.01	dBW	18
Tx antenna gain	2.15	dBi	19
EIRP	15.16	dBW	20
Tx line/waveguide loss	0.70	dB	21
Tx pointing loss	2.15	dB	22
Radiated power	12.31	dBW	23
PATH PARAMETERS			
Space loss	175.24	dB	24
Atmospheric losses	0.10	dB	25
Polarization losses	0.10	dB	26
Total path losses	175.44	dB	27
RECEIVER PARAMETERS			
Rx pointing loss	2.15	dB	28
Rx antenna gain	2.15	dBi	29
Rx system noise temperature	28.45	dB-K	30
Rx G/T	-26.30	dB-K	31
POWER SUMMARY			
Boltzmann constant	-228.60	dBJ-K	32
Noise spectral density (No)	-200.15	dBW-Hz	33
Received C/No	37.02	dB-Hz	34
Carrier suppression loss	0.00	dB	35
Received data C/No	37.02	dB-Hz	36
Data rate	29.29	dB-bps	37
Received Eb/No	7.72	dB	38
CODING AND BANDWIDTH SUMMARY			
Symbol per modulated symbol	1	-	39
Coded symbol rate	1.70E+03	sps	40
Analog signal BW	0.00	MHz	41
Available BW	6.00	MHz	42
BW occupancy	0.04%	-	43
Assumed coding gap	1.31	dB	44
DATA RATE SUMMARY (User inputs)			
AWGN capacity threshold	0.19	dB	45
Required Eb/No	1.50	dB	46
Rx Implementation Loss	2.00	dB	47
MA Loss	1.20	dB	48
Eb/No margin	3.03	dB	49

	Value	Units	Id
SPECTRUM CONFIGURATION			
Mission category	A	-	1
Frequency band	S	-	2
Network support	NTIA/FCC	-	3
Support service	Any	-	4
Carrier frequency	2.30E+03	MHz	5
Carrier wavelength	1.30E-01	m	6
Max. usable bandwidth	6.00	MHz	7
LINK CONFIGURATION			
Receiver type	Custom	-	8
Modulation	BPSK	-	9
Code rate	1/2	-	10
Baseband encoding	NRZ	-	11
Pulse shaping	SRRC	-	12
Shaping expansion factor	1.25	-	13
Modulation index	90.00	deg	14
Link distance	6000.00	km	15
User data rate	1.00E+03	bps	16
Desired link margin	3.00	dB	17
TRANSMITTER PARAMETERS			
Tx power	13.01	dBW	18
Tx antenna gain	2.15	dBi	19
EIRP	15.16	dBW	20
Tx line/waveguide loss	0.70	dB	21
Tx pointing loss	2.15	dB	22
Radiated power	12.31	dBW	23
PATH PARAMETERS			
Space loss	175.24	dB	24
Atmospheric losses	0.10	dB	25
Polarization losses	0.10	dB	26
Total path losses	175.44	dB	27
RECEIVER PARAMETERS			
Rx pointing loss	2.15	dB	28
Rx antenna gain	2.15	dBi	29
Rx system noise temperature	28.45	dB-K	30
Rx G/T	-26.30	dB-K	31
POWER SUMMARY			
Boltzmann constant	-228.60	dBJ-K	32
Noise spectral density (No)	-200.15	dBW-Hz	33
Received C/No	37.02	dB-Hz	34
Carrier suppression loss	0.00	dB	35
Received data C/No	37.02	dB-Hz	36
Data rate	30.00	dB-bps	37
Received Eb/No	7.02	dB	38
CODING AND BANDWIDTH SUMMARY			
Symbol per modulated symbol	1	-	39
Coded symbol rate	2.00E+03	sps	40
Analog signal BW	0.00	MHz	41
Available BW	6.00	MHz	42
BW occupancy	0.04%	-	43
Assumed coding gap	1.31	dB	44
DATA RATE SUMMARY (User inputs)			
AWGN capacity threshold	0.19	dB	45
Required Eb/No	1.50	dB	46
Rx Implementation Loss	2.00	dB	47
MA Loss	0.00	dB	48
Eb/No margin	3.52	dB	49

Figure 7: Comparison of link budgets using CDMA (left) and MF-FDMA (right)

MF-FDMA, despite being the conceptually simplest approach, was considered to be the best alternative given a 6 MHz bandwidth allocation at S-band. Figure 6 shows the frequency plan for MF-FDMA. The initial 6 MHz are first divided into three 2 MHz partitions, one for each orbital plane. Each of these 2 MHz is then split into four 500 kHz channels that are used to transmit by two satellites 180 deg apart within the orbital plane (i.e., we use frequency reuse to minimize bandwidth requirements). Since all transmissions use BPSK and SRRC pulse shaping with 0.25 roll-off factor, the maximum symbol rate is 200 ksps. Since the maximum supportable data rate due to power constraints and encoding is 1 kbps with Turbo ½ encoding, the maximum required symbol rate is limited to 2 ksps. Finally, each spacecraft transmits at the center frequency of the channel it has been assigned. At the receiver, an upper bound on the experienced Doppler shift is approximately ±124 kHz (corresponding to two satellites traveling in opposite directions

at 7.8km/s) which, therefore, fits well within the 500 kHz of the allocated channel. Four channels per plane are needed for all 8 satellites because we can re-use frequency for antipolar nodes. Each satellite radio must be able to "tune into" (thus requiring separate tracking loops) are 2 channels for intra-plane (left and right one) and 4 channels per plane for inter-plane, thus 8 channels total.

The MF-FDMA system thus avoids interference without any data rate penalty. Furthermore, full-duplex transmission can potentially be achieved by carefully planning transmission over the 50% bandwidth available within each channel. It also ensures that no network packets are dropped due to collisions at the physical layer. This, however, poses a problem at the network layer: Packets transmitted by node A towards node B (next hop) are possibly also received by other nodes. These, in turn, do not necessarily know that they have eavesdropped into the conversation between A and B, and thus re-forward the packet creating an exponential explosion of traffic. To avoid this problem, we assume that all bundles in the network carry an extension block that identifies the intended next hop of any given packet (as opposed to the bundle header, which has the address of its final destination) as described in the previous section. If, upon receipt of a bundle, the content of this extension block is not the present node's ID, its processing is immediately stopped. The viability of the MF-FDMA scheme assumes that about 3MHz of bandwidth (or 3.5% of TDRS' allocation) can be secured for the constellation, i.e., not shareable with other missions. Figure 7 compares the link budgets of both approaches. Ultimately, the CDMA trades are worth exploring only if no bandwidth for MF-FDMA can be secured.

C. Network Layer Simulation

DTN was 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 [31], the Licklider Transmission Protocol [32], or the Schedule Aware Bundle Routing [33]), 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 implemented in Python using Simpy [34], a discrete-event engine built upon the concept of coroutines (or asynchronous functions in the latest Python versions).

The network layer simulation assumes each satellite to be a DTN-enabled node with a simplified version of the Bundle Protocol and the Schedule Aware Bundle Routing Protocol. Packets are routed doing a shortest path search over the time-varying network topology. The objective is to minimize the best case packet delivery time. In the real DTN routing protocol specification, an ad-hoc heuristic algorithm to predict congestion in links based on local information is specified. This heuristic is partially the reason why routing in DTN is limited to ~15 nodes since it renders the routing procedures computationally complex. For the baseline 24-sat constellation in this paper, a simplified version of the heuristic is utilized to minimize computational burden. The route followed to destination will have no loops, but no effort to minimize number of hops is made.

The DTN simulation uses the two inputs from the OM module: The 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. Additional assumptions include:

- Satellites can exchange information whenever there is visibility between them with a nominal distance of 6000km for propagation delay calculations.
- Nominal link data rate of 1kbps is fixed and constant during all simulations. Scalability with data rate is also tested by performing simulation up to 1Mbps (see Figure 12)
- Link data rate is effective rate after losses and protocol overhead (not accounted for)

The output performance metrics are the number/Percentage of packets lost (because they are un-routable and cannot be delivered before the end of the simulation) and the latency experience per packet in time units.

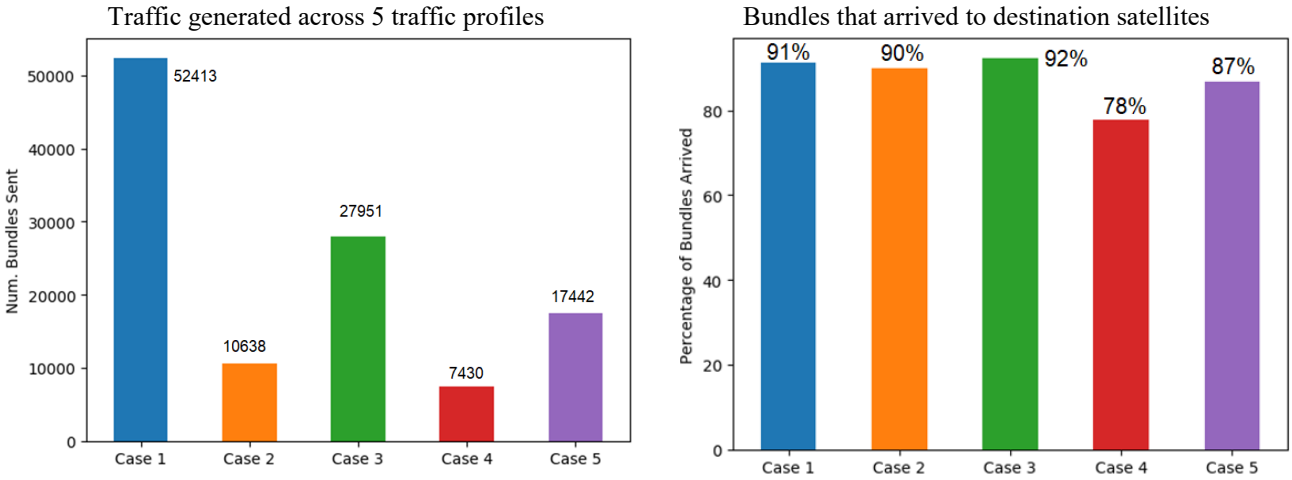

Figure 8: Packets delivered (right) as a function of number of packets generated (left)

A new packet or bundle is generated every time an observation is taken by a satellite. It has an associated destination satellite and priority. Each packet is assumed to contain 1645 bits of data, representing observed latitude, longitude, time, and parameters or observed data for the flood scenario. An additional (pessimistic) 20% overhead is assumed for protocol headers/trailers. Consequently, the bundle size is assumed to be 2 kbit. Figure 8 left shows the number of packets generated over a 6 hour simulation in 5 case studies of floods across 40 cities worldwide (1), 10 cities located close to each other (2) and sparsely worldwide (3), and 5 cities located closely (4) and sparsely (5). In spite of the very different traffic profiles, DTN is able to route most of the packets to their destination (Figure 9-right). Bundle drops occur due to bundle errors (corruption is not considered in this study), expiry (bundle arrives to node after its TTL has expired), unroutable (router cannot find a valid path to destination because all routes are full or because the contact plan is finite) or discarded (because of the extension block value). In practice, if >90% of all bundles reach destination, then the missing ones are typically stuck in the system because the contact plan is finite. Therefore, we consider such a scenario to be no loss.

The end-to-end latency experienced by 8341 2kbit bundles generated by all satellites and sent over a 6 hour simulation were statistically compared to the gaps in access by the satellite constellation on any region of interest - Figure 9. This simulation, run in less than 1 minute typically, showed that the latency of bundle transfer increased non-monotonically with prioritization between satellite pairs. 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 re-routed 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 [31]. 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. While the extension block mechanism is standardized and should be supported by all DTN compliant implementations, the utilization of the extension block as in the CGR routing is not a standard feature.

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 up to 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.

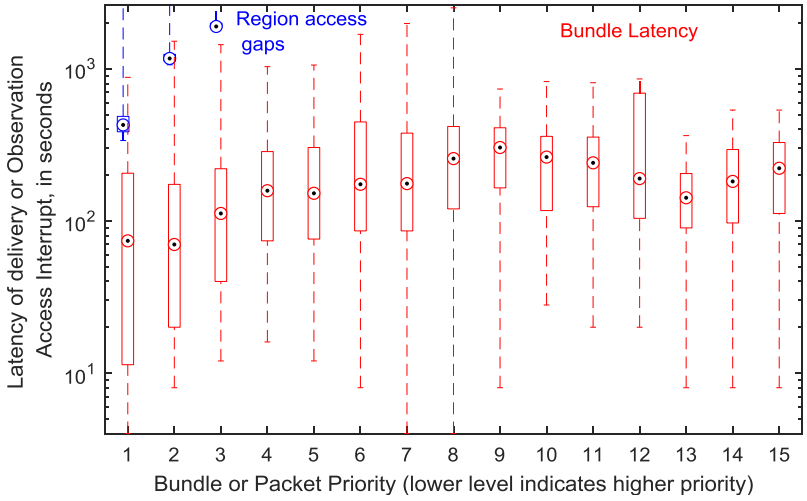

Figure 9: Latency of data bundle delivery over all satellite pairs compared to the gaps between satellite FOR access to any region. If longest latency is less than shortest gap, for pairs with the same priority, each satellite can be considered fully updated with information from all others and perfect 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.

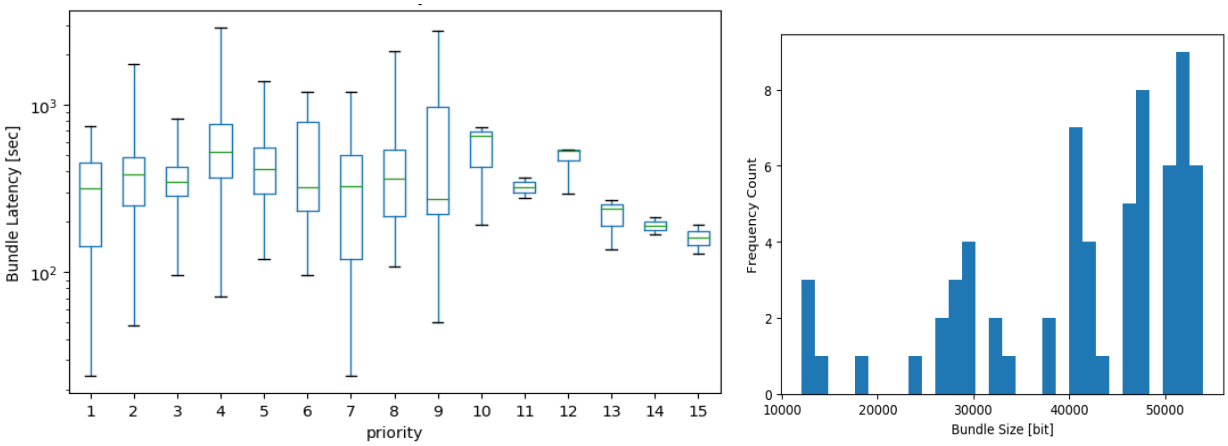

Figure 10: Latency of data bundle delivery (left) over all satellite pairs using the aggregation approach (right) results in less variance but larger latency compared to Figure 9 – red error bars.

An alternative approach was also explored where data from all observations by a satellite over a region, i.e. multiple observation points, is aggregated into a single DTN bundle per satellite destination and priority. Assuming a 2 kbit bundle per observation, the histogram in Figure 10-right shows how likely it is to send an aggregated bundle of given

bundle size. Bundles up to 60 kbits were generated, a reasonable size in network operations. In the 6-hour simulation of Figure 8, instead of 17442 bundles sent in Case 5, the aggregate approach causes 407 bundles to be sent, 83% of which are delivered using a 1kbps link, 99% using a 10kbps link and 100% for 100kps or higher links. Therefore, results showed little difference in successful delivery probability compared to the single packet approach, which was always around 85% (note that 100% probability is not possible since bundles generated just prior to terminating the simulation are counted as lost). As for latency as seen in Figure 10-left, typical values of 100 to 500 seconds were observed, with worse case packets being delivered as late as half an hour after generation. Less variance was observed because fewer number of packets were sent and larger latency was also induced by larger files being sent (i.e., not an entirely fair comparison).

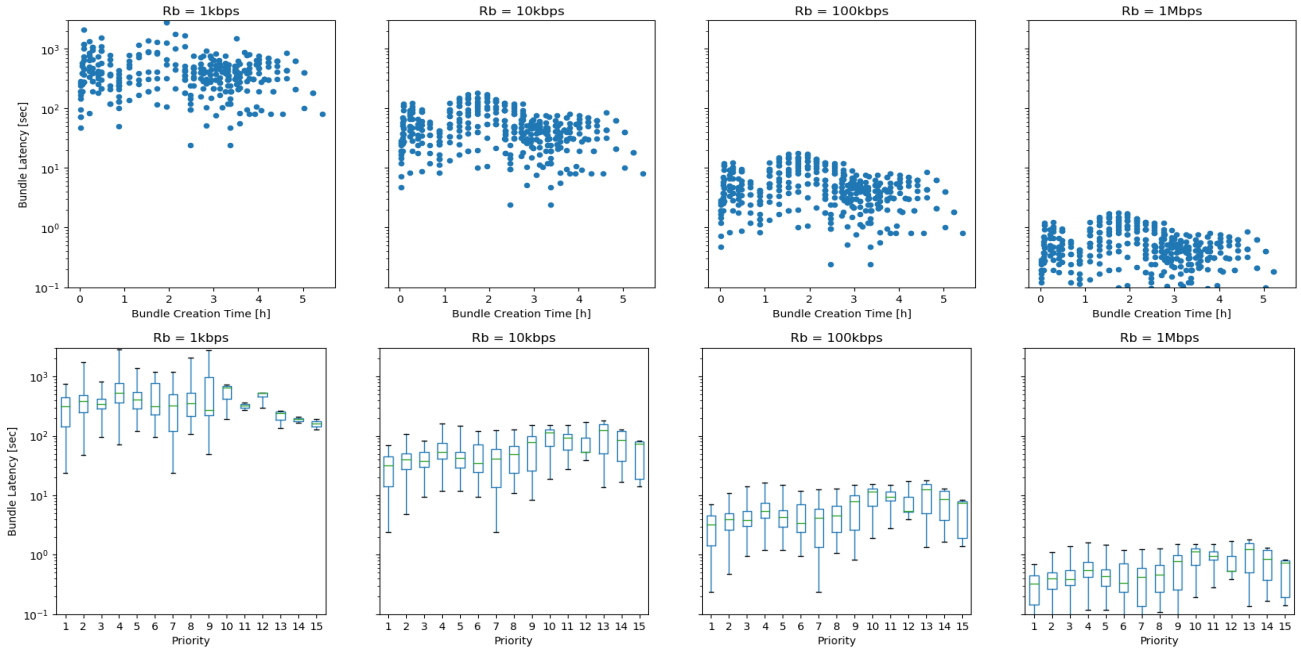

Figure 11: Bundle latency as a function of creation time (top), priority (bottom), data rate (left to right panels)

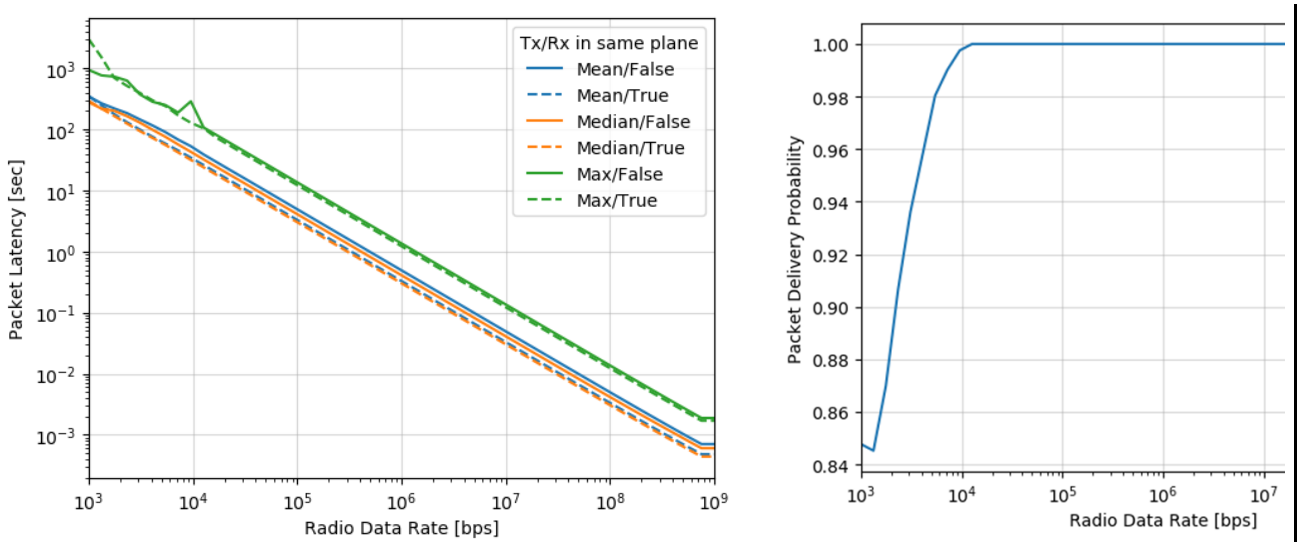

Figure 12: Dependence of packet or bundle latency (left) and delivery probability (right) on link data rate. Only Case 5's traffic profile from Figure 8 is considered for this simulation.

Latency was quantified as a function of bundle creation time, priority and increasing the link rate by one, two and three orders of magnitude (from the initial 1 kbps), as seen in Figure 11. The plots on creation time dependence over 6 hours confirm that the network is always in a stable operating point. If the network is operating in an unstable point of operation (i.e., too congested), bundles created later in the simulation should experience more latency as buffers in all DTN nodes are increasingly full. Within stable operations, it is confirmed that each order of magnitude increase in the data rate corresponds to an order of magnitude decrease in bundle delivery latency - Figure 11 (bottom) aggregated as statistics in Figure 12-left. For the purpose of system trades such as dependency on power budgets or coupling observation-crosslink schedules, the statistics of packet latency may be modeled using the following regression model as fit to the results in Figure 12-left.

$$\log(medianLatency) = -0.99 * \log R_b + 5.46 \ ; \ R^2 = 0.9998$$
$$\log(meanLatency) = -0.99 * \log R_b + 5.53 \ ; \ R^2 = 0.9998$$
$$\log(maxLatency) = -1.01 * \log R_b + 6.02 \ ; \ R^2 = 0.9985$$

Packets going across planes experience more latency, but the difference is at most 50 seconds in worst case (insignificant given 100min TTL). Figure 12-right confirms that almost 99% delivery probability was achieved with just 10 kbps, while average packet latency was shown to be inversely proportional to link rate. Inter-plane packet delivery suffers more than intra-plane. Note that in Figure 8, we only account for latency due to queuing or transmission of bundles. Additional latencies incurred while processing a bundle (e.g., computing a routing solution) are considered negligible.

V. DTN Performance across Use Case Variations

The previous section showed that inter-sat links for constellation coordination are supported by current technology. From a physical layer perspective, it seems possible to deploy a constellation with inter and intra-plane links at 1kbps over a wide range of distances. At the data link layer, it is recommended that constellations secure 3 MHz of BW at S-band to operate. This ensures that interference can be fully mitigated by frequency multiplexing. While the performance of the network layer is acceptable with links at 1kbps, much better performance is expected if the average data rate is increased by 10x. In the future, we will consider (1) alternative implementations to the physical layer that enable this 10x improvement (e.g., variable data rate), and (2) reducing link range and contact times.

We compared the performance of onboard scheduling with DTN links enabled (at planning horizons of 5 to 15 minutes with re-planning every 3 to 10 minutes) against a ground based implementation (planning every 99 to 198 minutes, corresponding to 7 to 4 ground station contacts in the 6-hr simulation). While the scheduler uses a greedy dynamic algorithm[2] applied to the objective function of maximizing flood magnitudes observed based on past precipitation measurements, they primarily serve as representative examples for the utility of DTN-enabled communication. The communications framework presented in this paper could be easily applied to other schedulers or objective functions, as determined by the science relevancy scenario. Results for the different use cases are summarized in Table 2. The best cases are bold-faced and every other row is compared to the best case as a % in brackets. If re-planning is not considered, longer planning horizons (~198 minutes, i.e. once in two orbits) yield the maximum value per observation (0.591); however this approach misses out on flood evolution therefore does not do well in terms of total value across all observations (total flood magnitude is ~6.5% lower). When the scheduler runs on onboard, re-planning frequency is constrained by onboard power or constraints on processing the bundles streaming in through DTN. For our fast-changing precipitation-causing-flood case, frequent replans for shorter horizons (just as shorter DTN latencies) improve total and per unit flood magnitude because plans are based on more updated values. However, observational value plateaus out at 5-min replans as negative effects of shorter planning horizons balance the diminishing returns of more frequent replans.

When the scheduler runs on the ground where all GS are assumed immediately synchronized, there is no risk of overlapping observations (if DTN bundles don't reach in time) and all satellites know each other's centralized schedules. Even when each satellite gets 2 GS contacts per orbit (i.e. 30/day), value updates are from at least an orbit earlier due to collection-uplink-reschedule-downlink latency between any satellite pair and the total and unit flood magnitude observed is ~3% less than any DTN-enabled decentralized run. Satellites with 1 GS contact per 99min orbit (15/day) are ~7% worse. For context, current small satellite missions commit to 2 contacts per day at NASA and 4-5 per day commercially; so the benchmarked scenario already implies a significant leap above current GS infrastructure. Onboard implementations will show higher (or lower) performance improvement over ground for phenomena that

evolve more, e.g. wildfire spreading (or less, e.g. daily snow melt) rapidly, for orbits that are more inclined to the Equator, and for regions of interest that are more equatorial (than the poles). Both implementations observe ~98% more flood magnitude in 6hrs than a non-agile constellation with its radars pointed in any static direction (no re-orientation).

Table 2: Comparison of schedulers run Onboard with DTN vs. in Ground Stations (GS). The centralized planner's horizon is assumed to be the time between GS contacts (99-198 min). Total flood magnitude observed is the best for DTN-enabled onboard scheduling (2703.7) compared to ground-based scheduling (2528.5). In comparison, the same constellation with non-agile satellites which can point their radars in a static direction only (e.g. nadir) observe a total flood magnitude of 50.16 i.e. ~98% lower than the DTN-enabled case.

	Planning Horizon (mins)	Total Observed Flood Magnitude	Flood Magnitude per Observation	Flood Magnitude per Observation (w/o replanning)	Max Time for each planHorizon (mins)	Max Time per unit planHorizon
Decen-tralized Plan Onboard before entering Region	5 (3m replan)	2661.7 (-1.6%)	0.825 (-0.7%)	0.1 (-83.1%)	**1.794 (best)**	0.359 (+1336%)
	10 (5m replan)	**2703.7 (best)**	**0.831 (best)**	0.1 (-83.1%)	2.374 (+32.3%)	0.237 (+848%)
	15 (10m replan)	2638 (-2.43%)	0.819 (-1.44%)	0.1 (-83.1%)	2.374 (+32.3)	0.158 (+532%)
Central-ized Plan on Ground	GS contact every 50m	2580.3 (-4.56%)	0.799 (-3.85%)	0.156 (-73.6%)	3.705 (+107%)	0.037 (+48%)
	GS contact every 99m	2528.5 (-6.5%)	0.787 (-5.3%)	**0.591 (best)**	5.029 (+180%)	**0.025 (best)**

The key metric to quantify DTNs performance in the content of spacecraft coordination is bundle delivery latency and percentage of packets delivered. A future publication[28] will show how performance varies as a function of constellation topology, EO observation targets and density, and spacecraft specifications. The input variables considered are: bundle size (between 1kbit and 25 kbits), number of packets generated per region (as a function of instrument duty cycle and steering capability), number and distribution of regions to observe (the baseline case of 5 regions will be extended to 10 and then 40 regions globally per the 5 cases), constellation structure (baseline case will be extended to 3 planes with 10 satellites each i.e. 30 satellites in total, 4 planes with 8 satellites each i.e. 32 satellites in total, 2 planes with 10 satellites each i.e. 20 satellites in total), TTL so that the network does not flood with redundant bundles, and power of the spacecraft radio and therefore data rate (between 1 kbps and 100 kbps). The last variable influences data relief to the network while all the others influence data load on the network.

For any architecture, the goal is to first ensure that it is stable, i.e. the maximum number of bundles per satellite does not grow continuously. Since data collection is restricted to access intervals to ROIs, it represents a dense data load but is sporadic; data at any satellite is expected to rise during such collections but expected to disseminate via the DTN slowly over time. If the data rate does not support the rise of data collected, the bundle size, packet number, ROIs and number of satellites will need to be scaled down to ensure stability. We then compare bundle transfer latencies across all satellite pairs, weighted by priority, for all the stable architectures. The tradespace will let us architect the constraints on payload duty cycle, supportable regions and constellation size as a function of communication capability, and therefore design well-coordinated constellations.

Acknowledgments

This project has been funded and supported by NASA's New Investigator Program, Earth Science Technology Office, and the Interplanetary Network Directorate at the Jet Propulsion Laboratory. We have released the DTN simulation as open source software. It is available for download at https://github.com/msancheznet/dtnsim, with a reference guide at https://dtnsim.readthedocs.io/en/latest/API.html#module-bin.main

References

[1] D. Mandl, S. W. Frye, M. D. Goldberg, S. Habib, and S. Talabac, "Sensor webs: Where they are today and what are the future needs?," in *Dependability and Security in Sensor Networks and Systems, 2006. DSSNS 2006. Second IEEE Workshop on*, 2006, pp. 65–70, Accessed: May 28, 2017. [Online]. Available: http://ieeexplore.ieee.org/abstract/document/1630363/.

[2] S. Nag *et al.*, "Autonomous Scheduling of Agile Spacecraft Constellations with Delay Tolerant Networking for Reactive Imaging," presented at the International Conference on Planning and Scheduling (ICAPS) SPARK Applications Workshop, Berkeley, California, U.S.A., 2019.

[3] M. J. Barnsley, J. J. Settle, M. A. Cutter, D. R. Lobb, and F. Teston, "The PROBA/CHRIS mission: A low-cost smallsat for hyperspectral multiangle observations of the earth surface and atmosphere," *Geosci. Remote Sens. IEEE Trans. On*, vol. 42, no. 7, pp. 1512–1520, 2004.

[4] J. Doubleday *et al.*, "Autonomy for remote sensing — Experiences from the IPEX CubeSat," in *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Jul. 2015, pp. 5308–5311, doi: 10.1109/IGARSS.2015.7327033.

[5] J. Hanson *et al.*, "Nodes: A flight demonstration of networked spacecraft command and control," presented at the Small Satellite Conference, Logan, Utah, 2016.

[6] C. Boshuizen, J. Mason, P. Klupar, and S. Spanhake, "Results from the Planet Labs Flock Constellation," presented at the AIAA/USU Small Satellite Conference, Logan, Utah, 2014, Accessed: Aug. 18, 2014. [Online]. Available: http://digitalcommons.usu.edu/smallsat/2014/PrivEnd/1/.

[7] S. Augenstein, A. Estanislao, E. Guere, and S. Blaes, "Optimal scheduling of a constellation of earth-imaging satellites, for maximal data throughput and efficient human management," 2016.

[8] E. Shao *et al.*, "Area Coverage Planning with 3-axis Steerable, 2D Framing Sensors," Delft, The Netherlands, 2018.

[9] D.-H. Cho, J.-H. Kim, H.-L. Choi, and J. Ahn, "Optimization-Based Scheduling Method for Agile Earth-Observing Satellite Constellation," *J. Aerosp. Inf. Syst.*, vol. 15, no. 11, pp. 611–626, 2018, doi: 10.2514/1.I010620.

[10] L. He, X. Liu, G. Laporte, Y. Chen, and Y. Chen, "An improved adaptive large neighborhood search algorithm for multiple agile satellites scheduling," *Comput. Oper. Res.*, vol. 100, pp. 12–25, 2018.

[11] S. Nag, J. L. Rios, D. Gerhardt, and C. Pham, "CubeSat constellation design for air traffic monitoring," *Acta Astronaut.*, vol. 128, pp. 180–193, 2016.

[12] V. Ravindra, S. Nag, and A. Li, "Ensemble-Guided Tropical Cyclone Track Forecasting for Optimal Satellite Remote Sensing," *IEEE Trans. Geosci. Remote Sens.*, 2020.

[13] S. Nag, C. K. Gatebe, and T. Hilker, "Simulation of Multiangular Remote Sensing Products Using Small Satellite Formations," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 2, pp. 638–653, 2017.

[14] S. Nag, T. Hewagama, G. Georgiev, B. Pasquale, S. Aslam, and C. K. Gatebe, "Multispectral Snapshot Imagers onboard Small Satellite Formations for Multi-Angular Remote Sensing," *IEEE Sens. J.*, vol. 17, no. 16, pp. 5252–5268, 2017, doi: 10.1109/JSEN.2017.2717384.

[15] J. Vander Hook *et al.*, "Mars On-Site Shared Analytics Information and Computing," in *Proceedings of the International Conference on Automated Planning and Scheduling*, 2019, vol. 29, no. 1, pp. 707–715.

[16] L. Jian and W. Cheng, "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*, 2008, pp. 199–203, Accessed: May 30, 2017. [Online]. Available: http://ieeexplore.ieee.org/abstract/document/4630799/.

[17] B. G. Holden, "Onboard distributed replanning for crosslinked small satellite constellations," SM thesis, Massachusetts Institute of Technology, Cambridge, U.S.A., 2019.

[18] H.-L. Choi, L. Brunet, and J. P. How, "Consensus-based decentralized auctions for robust task allocation," *IEEE Trans. Robot.*, vol. 25, no. 4, pp. 912–926, 2009.

[19] X. Gallud and D. Selva, "Agent-based simulation framework and consensus algorithm for observing systems with adaptive modularity," *Syst. Eng.*, vol. 21, no. 5, pp. 432–454, 2018.

[20] J. van der Horst and J. Noble, "Task allocation in networks of satellites with Keplerian dynamics," *Acta Futura*, vol. 5, pp. 143–150, 2012.

[21] R. B. Linnabary, A. J. O'Brien, G. E. Smith, C. Ball, and J. T. Johnson, "Using Cognitive Communications to Increase the Operational Value of Collaborative Networks of Satellites," in *2019 IEEE Cognitive Communications for Aerospace Applications Workshop (CCAAW)*, 2019, pp. 1–6.

[22] R. Linnabary, A. O'Brien, G. E. Smith, C. Ball, and J. T. Johnson, "Open Source Software For Simulating Collaborative Networks Of Autonomous Adaptive Sensors," in *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*, 2019, pp. 5301–5304.
[23] V. Cerf *et al.*, "Delay-tolerant networking architecture," 2007.
[24] C. Caini, H. Cruickshank, S. Farrell, and M. Marchese, "Delay-and disruption-tolerant networking (DTN): an alternative solution for future satellite networking applications," *Proc. IEEE*, vol. 99, no. 11, pp. 1980–1997, 2011.
[25] P. G. Madoery, J. A. Fraire, and J. M. Finochietto, "Analysis of Communication Strategies for Earth Observation Satellite Constellations," *IEEE Lat. Am. Trans.*, vol. 14, no. 6, pp. 2777–2782, 2016.
[26] S. Nag, A. Li, and J. Merrick, "Scheduling Algorithms for Rapid Imaging using Agile Cubesat Constellations," *COSPAR Adv. Space Res.*, vol. 61, no. 3, pp. 891–913, 2018.
[27] E. Sin, A. Li, V. Ravindra, and S. Nag, "Autonomous Attitude Control for Responsive Remote Sensing by Satellite Constellations," presented at the AIAA SciTech Forum, Nashville, TN, 2021.
[28] S. Nag, M. S. Net, A. S. Li, and V. Ravindra, "Disruption Tolerant Networks for Reactive Spacecraft Constellations," submitted 2021.
[29] E. Peral, S. Tanelli, Z. Haddad, O. Sy, G. Stephens, and E. Im, "Raincube: A proposed constellation of precipitation profiling radars in CubeSat," in *Geoscience and Remote Sensing Symposium (IGARSS), 2015 IEEE International*, 2015, pp. 1261–1264, [Online]. Available: http://ieeexplore.ieee.org/abstract/document/7326003/.
[30] M. Sanchez Net, I. del Portillo, B. Cameron, E. F. Crawley, and D. Selva, "Integrated tradespace analysis of space network architectures," *J. Aerosp. Inf. Syst.*, vol. 12, no. 8, pp. 564–578, 2015.
[31] K. Scott and S. Burleigh, "RFC 5050: Bundle protocol specification," *IRTF DTN Res. Group*, 2007.
[32] S. Farrell, M. Ramadas, and S. Burleigh, "Licklider transmission protocol-security extensions," 2008.
[33] P. D. R. Standard and P. R. Book, "Schedule-Aware Bundle Routing," 2018.
[34] N. Matloff, "Introduction to discrete-event simulation and the simpy language," *Davis CA Dept Comput. Sci. Univ. Calif. Davis Retrieved August*, vol. 2, no. 2009, pp. 1–33, 2008.