ADVANCING AUTONOMY IN DISTRIBUTED SPACE SYSTEMS: INSIGHTS FROM ON-ORBIT TESTING WITH THE STARLING 1.0 MISSION

Caleb Adams, Jeremy Frank, Sergei Gridnev, Michael Iatauro, Brian Kempa, Caroline Lassiter, Richard Levinson
Intelligent Systems Division
NASA Ames Research Center, Moffett Field, CA, USA

Abstract

Autonomous decision-making is crucial for enabling and enhancing mission effectiveness in Distributed Space Systems (DSS). Autonomy in decision-making is highlighted as a crucial factor for multi-spacecraft missions, enabling spacecraft to operate independently, reducing reliance on ground control. This capability is significant for future deep-space missions, where communication delays and limited data transmission capacity make traditional command and control approaches impractical. The Distributed Spacecraft Autonomy (DSA) experiment onboard the Starling 1.0 Mission focuses on a GPS Channel Selection Experiment, using ionospheric phenomena such as the Equatorial Ionization Anomaly and Polar Patches as a science proxy to demonstrate autonomy. The GPS Channel Selection Experiment utilizes a dual-band GPS receiver to estimate plasma density in the ionosphere. Explorative and exploitative channel selections are employed based on the nature of observed phenomena. The DSA Flight Software utilizes the Core Flight System (cFS) framework, ensuring compatibility with the Starling 1.0 flight mission software. This paper describes the current status of the experiment, including a review of the flight software, operations tools and processes, and a description of 'DSA Firsts' in demonstrating fully distributed autonomous operations.

1 INTRODUCTION to AUTONOMY in DISTRIBUTED SPACE SYSTEMS

Autonomous decision-making is crucial for enabling and enhancing mission effectiveness in Distributed Space Systems (DSS). Autonomy in decision-making is highlighted as a crucial factor for multi-spacecraft missions, enabling spacecraft to operate independently, reducing reliance on ground control. This capability is significant for future deep-space missions, where communication delays and limited data transmission capacity make traditional command and control approaches impractical. The Distributed Spacecraft Autonomy (DSA) project focuses on advancing autonomy in Distributed Space Systems (DSS) through *Distributed Resource and Task Management, Reactive Operations, Swarm Commanding,* and *Network Communications.* The following subsections provide an overview of each focus area and how they contribute to the advancement of autonomy in DSS:

Distributed Resource and Task Management enables each spacecraft to generate, communicate, and execute its own schedule based on mission objectives and available data. By adopting a decentralized approach, DSS can achieve a high level of flexibility and adaptability in managing resources and tasks. This paradigm shift in command and control methodologies enables more efficient utilization of available resources, and the ability to respond dynamically to changing mission requirements or unexpected events. This capability is particularly valuable in scenarios with limited communication bandwidth or latency constraints, where centralized scheduling would be impractical.

Reactive Operations is concerned with algorithms that optimize data collection strategies in real-time, enabling dynamic sensing and adjustments to operations based on evolving mission conditions. By leveraging reactive operations, a DSS can adapt to changing situations, enhance data collection efficiency, and improve overall mission performance. As the scale of a DSS increases, an autonomous system can react more efficiently and rapidly than human operators. This shift of responsibilities empowers the system to adapt to dynamic circumstances and optimize resource allocation, leading to improved mission outcomes and increased operational efficiency.

Swarm commanding enables operation of DSA with a single ground command. Collaboration with the Starling 1.0 mission played a significant role in the DSA project. The integration of human-swarm interaction capabilities through ground control software enables operators to command and interact with the spacecraft swarm (DSS) as a collective entity.

Finally, *Ad-hoc Network Communications* is a critical component in the advancement of autonomy in DSS. DSA focuses on developing a communication infrastructure that is scalable, robust, and automatically self-configuring. By ensuring efficient and reliable communication among the distributed spacecraft, DSS missions can effectively exchange information, share situational awareness, and support collaborative decision-making. DSA has partnered with the Starling 1.0 mission and Starling has provided the capabilities listed above. DSA contributes to this focus area by building middle-ware to take advantage of the lower level network stacks Starling has provided.

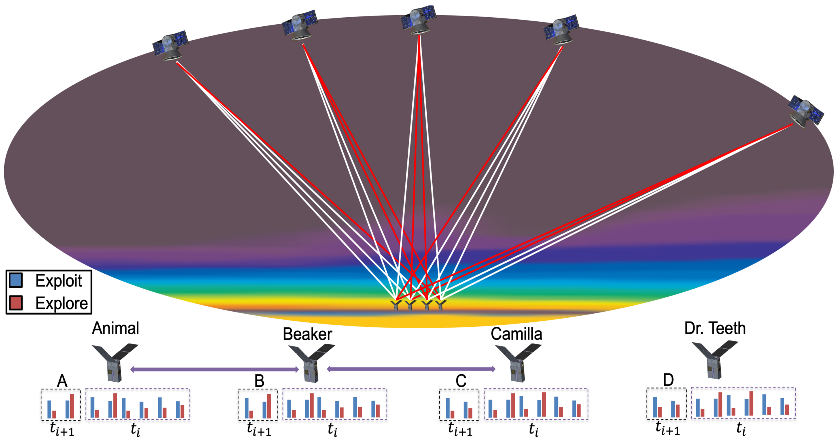

Figure 1: DSA Software performing autonomous GPS channel selection for TEC calculation. A time series below spacecraft A, B, C, and D represents the history of explore and exploit values used by the DSA software to autonomously select GPS channels; red and white lines of sight represented the variation in selected channels.

2 DSA and STARLING 1.0

DSA is a software payload on the Starling 1.0 mission [1]. The DSA-Starling flight demonstration centers around a GPS Channel Selection Experiment, described in detail in prior papers[2]. This experiment utilizes a dual-band GPS receiver to measure the total electron content (TEC) of the plasma between the spacecraft and GPS satellites. This experiment was selected as the primary demonstration due to its ability to showcase autonomous reconfiguration in response to natural phenomena without significant integration efforts or modifications to the spacecraft hardware.

2.1 GPS Channel Selection Experiment

The topside ionosphere is a transitional region between the ionosphere and the inner magnetosphere that displays many dynamic features. The GPS Channel Selection Experiment focuses on using a dual-band GPS receiver to estimate the plasma density in the ionosphere. By measuring the relative group delay between signals broadcast at different frequencies by GPS satellites, the receiver can capture a wide range of ionospheric phenomena. Two specific phenomena of interest, the Equatorial Plasma Bubbles [3] and Polar Patches [4], exhibit distinct behavior in TEC, and thus act as the features to be observed during the experiment. The experiment employs explorative channel selections (observe as many channels as possible) when the phenomena being

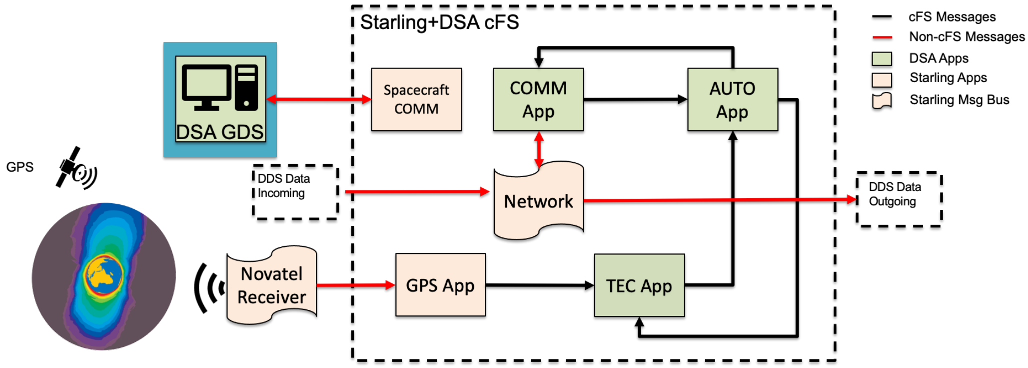

Figure 2: A simplified diagram of the 3 DSA applications operating within the Starling flight software environment, receiving GPS data, and communicating with the DSA Ground Data System (GDS), and utilizing the DDS network.

observed are large and homogeneous, and exploitative channel selections (focus observations on channels where the TEC count is highest) when the phenomena are spatially constrained and short-lived.

Figure 1 provides a simplified representation of a channel assignment scenario within the DSS, where multiple spacecraft receive signals from GPS satellites. The experiment constrains the number of channels each spacecraft can observe, requiring DSA to coordinate channel assignments across the DSS using shared sampling. In the case of spatially-constrained phenomena, simultaneous sampling allows multiple spacecraft to observe the features of interest from different vantage points. DSA performance will be evaluated based on their ability to match the optimal channel allocations and their responsiveness to changes in observed features and operating conditions.

2.2 Flight Software

The DSA Flight Software utilizes the Core Flight System (cFS) as the framework for each satellite's flight software. This choice ensures compatibility with the Starling-1 flight mission software. The DSA flight mission software consists of three apps within the (cFS) framework: the *COMM App*, *TEC App*, and *AUTO App*. The flight dataflow diagram (see figure 2) illustrates the flow of data from raw GPS instrument data to channel selections through the three cFS apps. The COMM App facilitates communication between the local autonomy software and other spacecraft, while the TEC App calculates relevant information from raw GPS range data. The AUTO App utilizes a Mixed Integer Linear Programming (MILP) solver to find optimal channel allocations by combining rewards from the TEC App and other spacecraft. We describe each app further below.

The purpose of the AUTO app is to decide what GPS channels each Starling spacecraft will monitor. As described in [5] we pose and solve the problem as a Mixed Integer Linear Program (MILP). Let S be the set of all Starling spacecraft s_i. Let s_i^c be each Starling spacecraft's channel capacity, the maximum number of GPS satellites it can observe. (Typically all s_i^c are identical; as discussed later, this is an artificial capacity constraint and can be configured operationally to explore different AUTO behavior.) Let s_i^v denote the set of GPS IDs visible to Starling spacecraft s_i. Let $g_j \in s_i^v$ denote GPS satellite with ID j. Let $r_{i,j}^e$ denote the explore reward for spacecraft s_i observing g_j; $r_{i,j}^e \propto \delta(s_i, g_j)$, the distance between the Starling satellite and the GPS satellite. Let $r_{i,j}^x$ denote the exploit reward for spacecraft s_i observing g_j. $r_{i,j}^x \propto TEC(s_i, g_j)$, the TEC reward for s_i monitoring g_j, obtained from the TEC app. Let r^c denote the coverage reward (a constant). Let $G = \cup_i s_i^v$, the set of all GPS satellites visible to at least one Starling spacecraft. Finally, $\alpha, \beta \in [0..1]$ are real-valued parameters controlling the blended combination of explore rewards, exploit rewards, and GPS coverage, corresponding to our requirements described in Sec 2.1.

The MILP has two types of binary $0, 1$ decision variables. Variable $o_{ij} = 1$ if and only if s_i is assigned to observe g_j. Assignments must obey the capacity constraints $\sum_j o_{i,j} \leq s_i^c$ for all $s_i \in S$. Variable $c_j = 1$ if and only if g_j is observed by at least b Starling spacecraft. At minimum, $b = 1$, meaning at least one Starling spacecraft observes g_j. This constraint translates to the following equivalence: $c_j = 1 \Leftrightarrow b \leq \sum_i o_{i,j}$ which can be modeled as linear constraints, leading to the following MILP:

$$\max \alpha \sum_{i,j} \beta r_{i,j}^e o_{i,j} + (1-\beta) r_{i,j}^x o_{i,j} + (1-\alpha) \sum_j r^c c_j \tag{1}$$

$$\text{s.t.} \sum_j o_{i,j} \leq s_i^c \qquad \forall s_i \in S \tag{2}$$

$$b \geq (\sum_i o_{i,j}) - M_1 c_j \qquad \forall g_j \in G \tag{3}$$

$$b \leq (\sum_i o_{i,j}) - 1 + M_2(1 - c_j) \qquad \forall g_j \in G \tag{4}$$

The AUTO App takes in the rewards from the TEC app, normalizes them, and gathers the reward states from the other instances of AUTO running on the other satellites and communicated over the COMM app. This *distributed consensus* approach aims to ensure each Starling spacecraft has the same information, and can solve the same problem, and obtain the same answer. Each instance of the AUTO app uses the rewards it received through the COMM app to generate the observation plan for the next tick.

Note that all of this information transmitted between Starling spacecraft *changes* the MILP from tick to tick; in particular, $r^e_{i,j}, r^x_{i,j}$ are recomputed from the rewards sent between Starling spacecraft. Furthermore, the sets s^v_i change as GPS satellites pass in and out of view. AUTO must therefore *regenerate* the MILP each tick before it is solved. The fact that the MILPs are constantly changing requires more general purpose automated reasoning capability than a dedicated and highly tuned solver.

AUTO uses lp_Solve as the underlying MILP solver, using a C++ wrapper around the lp_Solve's pure C API to facilitate model construction and solution extraction using cFS. AUTO gathers TEC rewards constructs the MILP model, and extracts the resulting plans.

The TEC App manages Total TEC reward information. The spacecraft collect observations from the onboard dual-band Novatel OEM719 GPS receiver already integrated into the Starling payload that provides precise orbit determination to the spacecraft in the DSS. The measurement used is TEC measured from the accumulated phase delay of the dual-band GPS signal as it passes through the plasma in the space between the transmitting GPS satellite and the receiver. Several forms of bias can complicate the process of comparing TEC signals received both from different GPS satellites and by receivers on different spacecraft in the DSS [6]. The largest of these are the Differential Biases that are imparted during the processing of the two signals in the hardware of the receiver. As described in [5], three different bias estimates were evaluated; we chose the Initial Value Subtraction, which is the Relative TEC with the first value in the timeseries used as the estimated bias correction. While Initial Value Subtraction does not accurately capture the absolute value of the signal, its relative simplicity, the preservation of the relative magnitude of the signal, and the fact that it can be applied to data as it is received makes it an attractive replacement for the absolute bias correction. Furthermore, for the purposes of the demonstration where feature recognition is more important than scientific accuracy, these sorts of datastream friendly corrections are sufficient.

As described in [5], the COMM App manages the inter-spacecraft crosslinks messaging and implements the Distributed Data Service (DDS) that allows satellites to communicate using a publish-subscribe framework. The COMM App receives messages from other cFS apps, translates them from cFS bus messages to DDS messages, sends them over the crosslink network, and translates received DDS messages back to cFS messages. The network implementation and the DDS layer on top is a joint effort between the Starling and DSA missions. DSA manages the application-level DDS topics created by the COMM Apps on all of the spacecraft to which these spacecraft subscribe and over which these spacecraft publish. DSA uses RTI's Micro DDS[7], which allows the management of certain quality of service (QoS) parameters. The Starling mission manages the transport protocol and communication network, which implements the Better Approach To Mobile Ad-hoc Networking (BATMAN) [8] protocol on top of the crosslink radio network.

3 HOW DSA OPERATES

DSA's high-level operational goals were set early in development: demonstrate a fully autonomous distributed space mission, and measure the benefits of autonomy for such systems. These were guiding principles through development and, indeed, were almost the only aspect of operations that persisted through flight. Even how the autonomy was controlled, what the benefits of the swarm commanding were, and how to measure them, had to change.

3.1 Early Mission Development

Starling 1.0's initial mission plan allocated one week of continuous operations to DSA, during which there would be frequent real-time contacts with each member of the swarm, lights-out downlink contacts, one month of analysis and planning, a second week of continuous operations, and finally one or two weeks of joint operations with the other experiments. We designed our demonstration to use swarm commanding to reconfigure all the spacecraft in real time with some limited monitoring of the autonomy, and use the downlink contacts to receive recorded telemetry of the autonomy's behavior.

3.2 Evolving Mission Operations

Subsequently, the Starling 1.0 mission assigned operations to the spacecraft vendor, Blue Canyon Technologies (BCT), which changed every aspect of the concept of operations. Real-time interaction was removed from the experiment plan entirely. This required that the measure of swarm commanding change from the count of ground commands required to change swarm configuration to the size of commanding products uplinked. BCT contracted with KSAT-Lite to provide the space-to-ground link, which created a tight scheduling constraint for DSA. DSA used Starling 1.0's crosslink radios to transmit telemetry and commands among the swarm, but a flight rule required that the crosslink radios be off during ground contacts to prevent any risk of coupling. The KSAT-Lite ground schedule was a rolling five-day schedule, updated daily; DSA needed to produce a fully grounded command plan at most five days in advance. BCT required one day to guarantee command products were uplinked before running on-board and Starling 1.0 required one day to verify those products before delivery. These restrictions, combined with a weekly operations cadence, resulted in a schedule of DSA running on-board Friday and Saturday; analyzing data, preparing new products, and verifying those products Monday and Tuesday; Starling 1.0 verifying the delivery on Wednesday; and BCT uplinking on Thursday.

This tight deadline motivated the development of the Mission Autonomy Demonstration Plan (MADP)—a series of scenarios consisting of application configuration tables and command sequences, each designed to collect the necessary data to verify a specific mission requirement or measure a specific flight metric, which the operations team was able to verify before launch using our containerized swarm environment. The intention of the MADP was to simplify decision-making during the brief analysis period. With it, analysis would consist of determining what requirements had been verified or not during the previous run and why. Planning would consist of determining what the next requirements to verify were or which needed re-work. Preparing command products would consist of assembling and re-scheduling the existing demonstration plans. The MADP also addressed requirements coverage by creating a traceability matrix to command sequences and parameter tables.

Further, a mission rehearsal in which DSA was requested to make a small alteration to our products—shifting the timelines by 15 minutes, proved to be extremely intensive to re-schedule; timelines exceeded 300 commands each across 4 spacecraft. This motivated the development of Splinter, a tool which characterizes command sequences in terms of their temporal constraints, both individually and across the swarm, and generates grounded command sequences. Splinter uses a Simple Temporal Network[9] implementation to propagate these temporal constraints and identify unsatisfiable constraints.

3.3 Post Launch Mission Operations

Launch is necessarily a major inflection point; it is the moment when, regardless of what a project's velocity was before, it abruptly increases to the 7.8 km/s required for low-Earth orbit. The first significant change post-launch was to the size of the swarm. One spacecraft suffered anomalies in both its ground radio and propulsion, leaving it operable, but too distant from the rest of the swarm to communicate via crosslink radios for months. While unfortunate, DSA was prepared to operate on a three-node swarm.

A second change was that bus, payload, and crosslink radio commissioning took longer than expected. The MADP was developed assuming an orderly and complete commissioning process before DSA began its own commissioning, but the team needed to engage before the flight characteristics of the payload and the crosslink radios (themselves an experimental component of Starling 1.0) were fully understood. Early experiments were assembled from the MADP scenarios, but analyses were directed at more fundamental questions or operability rather than which of the mission requirements had been satisfied. The MADP quickly stopped being useful for making tactical decisions, but remained critical in giving DSA a strategic view of how much remained to be done, how much time it would take, and how to integrate our schedule with the mission schedule.

Splinter proved critical in enabling the operations team to meet delivery deadlines and even be opportunistic about scheduling activities to help the Starling 1.0 team investigate issues, but command sequences are only one of the necessary products to deliver to run on-board. A complete delivery could consist of 34 files, more with additional AUTO app configuration tables, *per spacecraft*. Ensuring that the correct files got built and delivered to the correct spacecraft for a given experiment required both a system of procedures-as-checklists to track definition, construction, verification, and delivery, and git version control to coordinate development and maintain the delivery history.

Decision-making was facilitated by importing recorded telemetry into a PostgreSQL database, allowing the team to use the SQLAlchemy ORM to map the telemetry back to packet data structures for analysis in Jupyter notebooks and custom parsers to reconstruct AUTO behavior and infer state.

4 PRELIMINARY RESULTS

In this section, we describe preliminary analysis of downlinked data, and explain 'singular events' exhibiting DSA performance.

4.1 Network Report

DSA and Starling 1.0 have consistently established a 3-spacecraft cross-link network via our COMM application, as described in detail within figure 3. Our DDS network shares spacecraft state information and group messages. In figure 3 the gray areas represent time periods with no available data; darker gray represents times were data is missing for more than one spacecraft. (Continuous data does exist onboard the spacecraft, but it is not downlinked sequentially; often we must wait long periods of time before full data is available from EPs.) Additionally, the green vertical striped lines represent times when crosslink is turned on and the red vertical striped lines represent times when crosslink is turned off.

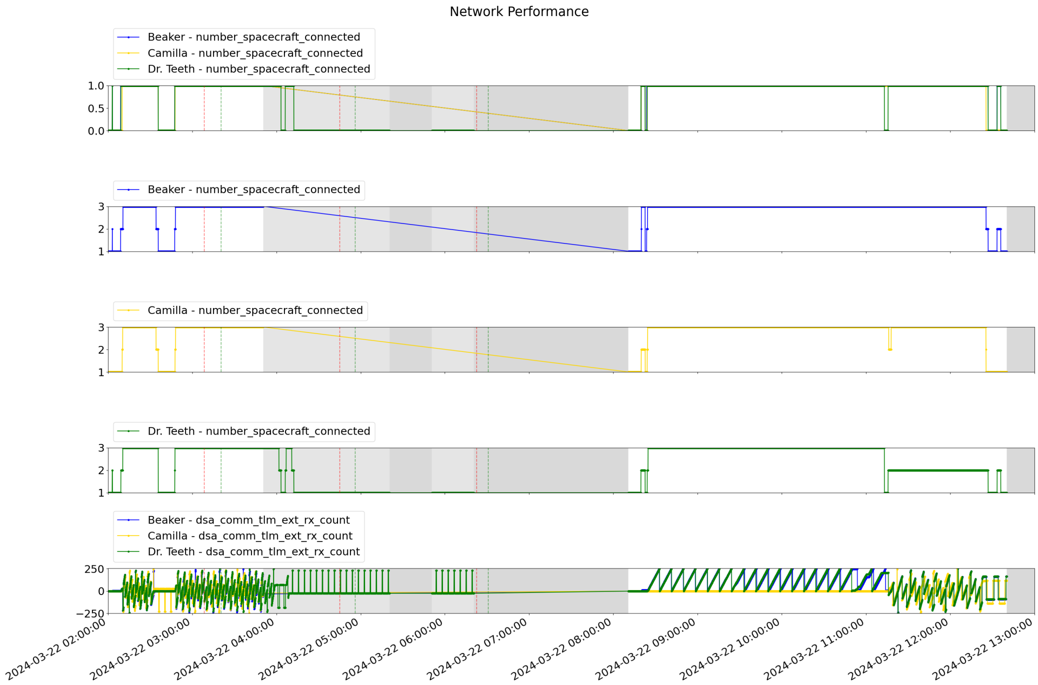

Figure 3: From top to bottom, over a select time period of EP8.1 day 1: Times when all spacecraft are connected to the network (1.0) and when a full network is not achieved (0.0); Number of spacecraft connected to Beaker, Camilla and Dr. Teeth; Accumulated telemetry packets from cross-link on Beaker, Camilla, and Dr. Teeth.

4.2 Consensus Report

Consensus exists when all of the spacecraft have the same plan. Periods of non-consensus are expected and can occur due to changes in network topology and configuration, GPS satellite visibility set s_i^v changes, changes in explore/exploit rewards $r_{i,j}^e, r_{i,j}^x$, or changes in the AUTO app configuration. Changes often occur within a time step that will cause non-consensus, as simulated in prior work [5]. We have analyzed the interval [01:14:57,02:50:38] in figure 3 (left). Consensus is achieved frequently (893 of 2709 ticks in this period) but often holds for only a few seconds. Figure 4 shows an example of consensus holding for 4 ticks, followed by lack of consensus. Notably, one of the plans generated on the non-consensus tick is 'invalid' in that a GPS satellite that is not in view is assigned (an issue corrected on the next tick).

SV	Time	SV2 plan	SV3 plan	SV4 plan
SV2	01:19:03	$(16, 17, 27, 30)$	$(8, 9, 14, 21)$	$(3, 4, 6, 7)$
SV3	01:19:03	$(16, 17, 27, 30)$	$(8, 9, 14, 21)$	$(3, 4, 6, 7)$
SV4	01:19:03	$(16, 17, 27, 30)$	$(8, 9, 14, 21)$	$(3, 4, 6, 7)$
...	...	...	...	...
SV2	01:19:06	$(16, 17, 27, 30)$	$(8, 9, 14, 21)$	$(3, 4, 6, 7)$
SV3	01:19:06	$(16, 17, 27, 30)$	$(8, 9, 14, 21)$	$(3, 4, 6, 7)$
SV4	01:19:06	$(16, 17, 27, 30)$	$(8, 9, 14, 21)$	$(3, 4, 6, 7)$
SV2	01:19:07	$(16, 17, 27, 30)$	$(8, 9, 14, 21)$	$(3, 4, 6, 7)$
SV3	01:19:07	$(16, 17, 27, 30)$	$(9, 14, 21, \mathbf{22})$	$(4, 6, 7, 8)$
SV4	01:19:07	$(16, 17, 27, 30)$	$(8, 9, 14, 21)$	$(3, 4, 6, 7)$

Figure 4: A sequence of 4 consecutive periods of consensus at 2024-03-22 01:19:03 - 01:19:06 followed by no consensus at 01:19:07 (indicated by bold font).

4.3 Coverage Report

Coverage requires every GPS satellites in view were selected by at least one Starling 1.0 spacecraft. Coverage is impossible when s_i^c (DSA channel capacity) is too small to cover s_i^v (GPS satellites visible to Starling spacecrafts s_i.). Coverage is achieved in 689 of the 893 ticks when consensus is achieved in the interval [01:14:57,02:50:38]. Figure 5 shows an instance of coverage.

SV	SV2 plan	SV3 plan	SV4 plan
SV2	$(17, 21, 27, 30)$	$(7, 8, 9, 14)$	$(2, 3, 4, 6)$
SV3	$(17, 21, 27, 30)$	$(7, 8, 9, 14)$	$(2, 3, 4, 6)$
SV4	$(17, 21, 27, 30)$	$(7, 8, 9, 14)$	$(2, 3, 4, 6)$

SV	Visible GPS Satellites
SV2	$(2, 3, 4, 6, 7, 8, 9, 14, 17, 21, 27, 30)$
SV3	$(2, 3, 4, 7, 8, 9, 14, 17, 21, 27)$
SV4	$(2, 3, 4, 6, 7, 8, 9, 14, 17, 21, 27, 30)$
All	$(2, 3, 4, 6, 7, 8, 9, 14, 17, 21, 27, 30)$

Figure 5: Full coverage and consensus at 2024-03-22 01:16:08; the GPS visibility sets are shown in the bottom of the table.

4.4 Latency Report

One definition of *Latency* is the difference between the time an event occurs that requires reconfiguration, and the time that consensus is achieved. The same events that lead to lack of consensus may lead to the need to reconfigure, and thus allow measurement of latency. However, it is possible that numerous events occur in rapid succession, and reconfiguration may not always be needed even when events do occur. Figure 6 shows an example of reactive operations that illustrates latency. In this case, multiple GPS satellites leave the visibility sets of DSA spacecraft at different times. However, after a 2-tick period of stability of GPS visibility sets, DSA is able to reach consensus.

4.5 DSA Firsts

Previous missions have demonstrated many ingredients of fully autonomous distributed space systems, e.g. space-to-space communications and command relay between multiple spacecraft [10], and onboard planning [11, 12] and reactive operations for a single spacecraft [13]. However, DSA-Starling represents the *first demonstration of a fully autonomous distributed space mission on* 3 *Starling 1.0 spacecraft.* Specifically, DSA-Starling accomplished the following:

- First fully distributed autonomous operation of multiple spacecraft.
- First use of space-to-space communications to autonomously share state information between multiple spacecraft.
- First demonstration of fully distributed reactive operations onboard multiple spacecraft.
- First use of fully distributed automated planning onboard multiple spacecraft.
- First use of a general purpose automated reasoning system onboard a spacecraft.

SV	Time	SV2 plan	SV3 plan	SV4 plan
SV2	01:14:58	(7, 21, 27, 30)	(7, 8, 9, 14)	(3, 4, 6, 22)
SV3	01:14:58	(17, 21, 27, 30)	(7, 8, 9, 14)	(3, 4, 6, 22)
SV4	01:14:58	(7, 21, 27, 30)	(7, 8, 9, 14)	(3, 4, 6, 22)
SV2	01:14:59	(7, 21, 27, 30)	(7, 8, 9, 14)	(3, 4, 6, 22)
SV3	01:14:59	(**17**,21, 27, 30)	(7, **9**,14, **22**)	(3, 4, 6, 8)
SV4	01:14:59	(7, 21, 27, 30)	(7, 8, 9, 14)	(3, 4, 6, 22)
SV2	01:15:00	(7, 21, 27, 30)	(8, 9, 14, 22)	(3, 4, 6, 7)
SV3	01:15:00	(7, 21, 27, 30)	(8, 9, 14, **17**)	(3, 4, 6, **22**)
SV4	01:15:00	(**17**,21, 27, 30)	(8, 9, 14, 22)	(3, 4, 6, 7)
SV2	01:15:01	(7, 21, 27, 30)	(8, 9, 14, **17**)	(3, 4, 6, 22)
SV3	01:15:01	(**17**,21, 27, 30)	(**7**,8, 9, 14)	(3, 4, 6, 22)
SV4	01:15:01	(7, 21, 27, 30)	(8, 9, 14, **17**)	(3, 4, 6, 22)
SV2	01:15:02	(7, 21, 27, 30)	(7, 8, 9, 14)	(3, 4, 6, 22)
SV3	01:15:02	(17, 21, 27, 30)	(7, 8, 9, 14)	(3, 4, 6, 22)
SV4	01:15:02	(7, 21, 27, 30)	(7, 8, 9, 14)	(3, 4, 6, 22)

SV	Time	Visible GPS Satellites
SV2	01:14:58	(2, 3, 4, 6, 7, 8, 9, 14, 17, 21, 27, 30)
SV3	01:14:58	(2, 3, 4, 6, 7, 8, 9, 14, 17, 21, 27)
SV4	01:14:58	(2, 3, 4, 6, 7, 8, 9, 14, 17, 21, 22, 27, 30)
SV2	01:14:59	(2, 3, 4, 6, 7, 8, 9, 14, 17, 21, 27, 30)
SV3	01:14:59	(2, 4, 8, 9, 14, 17, 21)(**3**, **6**, **7**, **27**)
SV4	01:14:59	(2, 3, 4, 6, 7, 8, 9, 14, 17, 21, 22, 27, 30)
SV2	01:15:00	(2, 3, 4, 6, 7, 8, 9, 14, 17, 21, 27, 30)
SV3	01:15:00	(2, **3**, 4, **7**, 8, 9, 14, 17, 21)
SV4	01:15:00	(2, 3, 4, 6, 7, 8, 9, 14, 17, 21, 22, 27, 30)
SV2	01:15:01	(2, 3, 4, 6, 7, 8, 9, 14, 17, 21, 27, 30)
SV3	01:15:01	(2, 3, 4, 7, 8, 9, 14, 17, 21, **27**)
SV4	01:15:01	(2, 3, 4, 6, 7, 8, 9, 14, 17, 21, 22, 27, 30)
SV2	01:15:02	(2, 3, 4, 6, 7, 8, 9, 14, 17, 21, 27, 30)
SV3	01:15:02	(2, 3, 4, 7, 8, 9, 14, 17, 21, 27)
SV4	01:15:02	(2, 3, 4, 6, 7, 8, 9, 14, 17, 21, 22, 27, 30)

Figure 6: Reactive operations during the period 2024-03-22 01:14:58 - 01:15:02. Consensus is lost at 2024-03-22 01:14:59, then achieved at 2024-03-22 01:15:02, as shown in the top half the table. GPS visibility sets are shown in the bottom half of the table; bold font shows newly visible GPS, the second parenthetical set indicates GPS satellites are no longer visible at 01:14:59. GPS satellite visibility is consistent from 01:15:01 - 01:15:02.

5 FUTURE DSA STARLING 1.0 OPERATIONS

The next steps for DSA operations include completing our planned experiments, including those designed to evaluate exploit behavior and demonstrate swarm commanding. Experiments involving all 4 Starling 1.0 spacecraft are possible as well. Once all of the experiment periods are complete, we will be able to more thoroughly evaluate DSA performance, and provide quantitative measurements of consensus, coverage, latency, and swarm commanding metrics. Many questions remain to be answered by these evaluations. For instance, how long is consensus achieved before it is lost? When is consensus lost, and why? Are there other measurements of consensus we can or should use? What is the correct measurement of latency, and what is the average latency in responding to events requiring reconfiguration? How does DSA performance differ based on configurations; i.e. is latency different in explore vs exploit configurations? We look forward to being able to report on the answers to these questions in subsequent publications.
Acknowledgments The authors of this paper would like to acknowledge Dr. Nicholas Cramer, who was the Project Manager of DSA during many formative moments. We would also like to thank NASA's Game Changing Development (GCD) program for their support over the years. Finally, we would like to thank NASA's Small Spacecraft Technology (SST) program for their support.

References

[1] Hugo Sanchez, Dawn M. McIntosh, Howard N. Cannon, Craig Pires, Joshua Sullivan, Simone D'Amico, and Brendan H. O'Connor. Starling1: Swarm Technology Demonstration. In *Proceedings of the AIAA/USU Conference on Small Satellites*, 2018.

[2] Daniel Cellucci, Nicholas Cramer, and Jeremy Frank. Distributed Spacecraft Autonomy. In *2020 AIAA Ascend Conference*. AIAA, 2020.

[3] Nanan Balan, LiBo Liu, and HuiJun Le. A Brief Review of Equatorial Ionization Anomaly and Ionospheric Irregularities. *Earth and Planetary Physics*, 2(4):257–275, 2018.

[4] Alex T Chartier, Cathryn N Mitchell, and Ethan S Miller. Annual Occurrence Rates of Ionospheric Polar Cap Patches Observed Using SWARM. *Journal of Geophysical Research: Space Physics*, 123(3):2327–2335, 2018.

[5] Nicholas Cramer, Daniel Cellucci, Caleb Adams, Adam Sweet, Mohammad Hejase, Jeremy Frank, Richard Levinson, Sergei Gridnev, and Lara Brown. Design and Testing of Autonomous Distributed Space Systems. In *Proceedings of the AIAA/USU Conference on Small Satellites*, 2021.

[6] M Noja, Claudia Stolle, Jaeheung Park, and Hermann Lühr. Long-term Analysis of Ionospheric Polar Patches Based on CHAMP TEC Data. *Radio Science*, 48(3):289–301, 2013.

[7] RTI Connext Micro. `https://www.rti.com/products/connext-dds-micro/`. Accessed 2022-03-18.

[8] batman-adv. `https://www.kernel.org/doc/html/latest/networking/batman-adv.html/`, 2008. Accessed 2022-03-18.

[9] R. Dechter, I. Meiri, and J. Pearl. Temporal constraint networks. *Artificial Intelligence*, 49:61–94, 1991.

[10] J. Hanson, A. Guarneros Luna, R. DeRosee, K. Oyadomari, J. Wolfe, W. Attai, and C. Prical. Nodes: A Flight Demonstration of Networked Spacecraft Command and Control. In *Proceedings of the Small Satellite Conference*, 2016.

[11] A. Jónsson, P. Morris, N. Muscettola, K. Rajan, and B. Smith. Planning in Interplanetary Space: Theory and Practice. In *Proceedings of the Fifth International Conference on Artificial Intelligence Planning and Scheduling*, 2000.

[12] D. Tran, S. Chien, R. Sherwood, R. Castaño, B. Cichy, A. Davies, and G. Rabbideau. The Autonomous Sciencecraft Experiment Onboard the EO-1 Spacecraft. In *Proceedings of the* 19^{th} *National Conference on Artificial Intelligence*, pages 1040 – 1045, 2004.

[13] F. Ip, J.M. Dohm, V.R. Baker, T. Doggett, A.G. Davies, R. Castaño, S. Chien, B. Cichy, R. Greeley, R. Sherwood, D. Tran, and G. Rabideau. Flood Detection and Monitoring with the Autonomous Sciencecraft Experiment Onboard EO-1. *Remote Sensing of Environment*, 101(4):463–481, 2006.