Cognitive Engine One: A Cross-Layer Framework for Autonomy in Multi-Provider Space Communications Environments

Adam Gannon*, Joseph Downey†, Rachel Dudukovich†, Aaron Smith†, Matthew Hilts†, Mick Koch†, Timothy Recker†, Ethan Schweinsberg† and Shadi Zogheib†
NASA Glenn Research Center, Cleveland, Ohio, 44135

David Brooks‡
SAIC, Inc., Reston, VA, 20190

Larry Vincent§ and Constantine Gemelas¶
ZIN Technologies, Middleberg Heights, Ohio, 44130

Janette Briones‖
NASA Glenn Research Center, Cleveland, Ohio, 44135

Spacecraft information needs, along with the communications environments in which data transfers occur, are becoming increasingly dynamic. Earth orbit is served by dozens of ground station and relay satellite providers – each with unique protocols and service capabilities. In this context, we present a system to automate end-to-end space communications across the protocol stack for each of several providers. The proposed system schedules contacts with providers, configures point-to-point RF links, and ensures data is transferred to its destination. Data monitoring and rescheduling provide an automated failover capability in the event of unsuccessful contacts. We verify system performance in a high-fidelity emulation testbed. Results from simulated mission use cases demonstrate the system reduces data latency, prevents exhaustion of onboard storage, and prioritizes transfer of urgent observations.

I. Introduction

Future science and exploration missions in Earth orbit and beyond will generate increasingly large data volumes and require progressively higher capacity from networks which support space communications [1]. Timely access to observation data with minimal latency can be critical for monitoring natural hazards [2]. Event-driven operations

*Engineer, Research and Engineering Directorate; Corresponding Author (adam.gannon@nasa.gov).
†Engineer, Research and Engineering Directorate.
‡Senior Electrical Engineer, Civil and Space Group.
§Software Engineer.
¶Work performed as a Software Engineer with ZIN Technologies, now with HX5, LLC.
‖Principal Investigator for Cognitive Communications, Research and Engineering Directorate.

like those used to capture transient phenomena make communication needs hard to anticipate in advance [3–5] and challenge traditional paradigms of allocating time on ground stations and relay satellites days or weeks in advance of contacts [6, 7]. Recent concepts envision constellations of small spacecraft providing high revisit rates over targets to gain understanding of natural processes [8]. These less-expensive spacecraft have the potential to make missions more cost-effective so long as the increase in satellites does not lead to a corresponding increase in operations cost.

At the same time, mission operators are leveraging new service offerings to communicate with their spacecraft [9, 10]. In the United States, the National Aeronautics and Space Agency (NASA) intends to phase out its government-operated ground stations and relay satellites in favor of a mix of commercially-provided services [11, 12]. Though future unified approaches are desirable, spacecraft wishing to interoperate between many providers currently face an array of different protocols, frequency bands, network configurations, and management interfaces across providers. Differences in orbits, antenna sizes, network configurations, and service models lead to large differences in throughput, latency, availability, and cost. While this heterogeneity presents an opportunity to well-match a mission's needs to a service, it also imparts a corresponding increase in complexity of managing communications for a spacecraft. Increased levels of autonomy across the protocol stack are necessary for dynamic missions to take full advantage of this new environment without significantly increased cost or operator burden.

In this work, we present and experimentally validate a complete integrated system allowing full automation of the mission spacecraft's communications process in a multi-provider environment: from service acquisition to link establishment, data flow, and transfer verification. Communications service is provisioned in response to a spacecraft's real-time needs and allocated through interaction with the scheduling systems of multiple providers. In contrast to current practice, this scheduling process uses machine-to-machine communications with minimal lead time. For each scheduled contact with a provider, the framework autonomously configures provider-specific protocols to ensure compatibility across the physical, data link, and network layers. Application-layer traffic (e.g., spacecraft telemetry and science data) is transferred to end users with delay/disruption tolerant protocols employing mechanisms for verified delivery. To evaulate the system, we created a spacecraft communications testbed [13] containing high-fidelity emulations of radio links during contacts. We refer to this system as Cognitive Engine One (CE-1), named in recognition that automation is the first step towards cognitive communications for space [14].

Simulation of normal as well as off-nominal operations in the testbed highlights the system's capability to recognize failed contacts and reschedule backup service to maximize likelihood of timely data delivery faster than possible in other systems which require human-in-the-loop operation. This level of automation reduces burden on Earth-orbiting mission operators when working in a multi-provider future.

We summarize our main contributions as follows:

1) A distributed framework to automate all aspects of space communications, with interfaces on Earth and onboard the user spacecraft.

2) An integrated system which automatically provisions space communications service in response to the real-time data transfer needs of the user spacecraft.
3) Software which configures protocols across all layers as a spacecraft transitions between providers, allowing end-to-end data flow without human intervention.
4) Evaluation of key system features in a high-fidelity testbed emulating several relevant mission scenarios.

Following a review of relevant literature in Section II, key features of the system design are discussed in Section III. Algorithmic developments for data monitoring and service scheduling are presented in Section IV. Section V describes the emulation testbed and its relevant design features. In Section VI, the system is evaluated on the testbed across several scenarios highlighting system handling of link failures, storage exhaustion, and arrival of unexpected data. Section VII concludes with discussion on avenues for improved decision making and learning which will be considered in future work.

II. Related Work

Many frameworks performing cross-layer management and optimization have been proposed for terrestrial applications [15–20]. Adaptation of these techniques to space is difficult due to unique conditions of the communications environment including: highly-directional links, frequency-separated uplink/downlink bands carrying asymmetrical data volumes, and other challenges. Furthermore, networks of ground stations and relay satellites frequently employ varied and sometimes incompatible standards across the protocol stack. These include those by the Consultative Committee for Space Data Systems (CCSDS), the European Telecommunications Standards Institute (ETSI), or custom standards developed by a provider for use in their network. Though fifth generation (5G) mobile broadband has future potential to harmonize space standards through inclusion of non-terrestrial network (NTN) provisions [21, 22], systems designed for interoperability must currently contend with a variety of protocols.

We focus the remainder of this section on a review of space-specific technologies with relevance to our work: i) establishment and adjustment of radio links, ii) routing within space networks, iii) scheduling communications services, and iv) integrated space communication systems and their evaluation. In each subsection, we briefly note open issues in the topic.

A. Physical-Layer Compatibility and Adaptation

As commercial communications offerings continue to expand, a single terminal capable of interoperating between dissimilar services is highly desirable. Benefits of integrating a suite of satellite communications protocols are discussed in [23] for terrestrial use and in [24, 25] for space-based users. Prototype spacecraft terminals capable of protocol compatibility with both government and commercial relay satellites have been demonstrated in over-the-air tests from the ground [26, 27] and will form the basis of the Polylingual Experimental Terminal (PExT) technology demonstration mission [28]. Prior work in [26] was primarily focused on physical-layer flexibility and relied on operator scripting to "roam" between providers. A more rigorous automation framework would better manage the complexity of configurations across many providers while reducing burden to mission operators.

After establishment, a link must remain robust to impairments including: precipitation, interference, and time-varying path loss. Adaptive coding and modulation (ACM) mechanisms (e.g., [29, 30]) provide real-time adjustment of data rates in response to changing link conditions. While widely used in fixed and mobile satellite services connecting terrestrial users, ACM has seen limited use in communication with spacecraft themselves. Experiments in the literature use a receiver controlled by the spacecraft operator connected to the satellite through bent-pipe relays [31] or mission-operated ground stations [32]. ACM with commercial services requires compatibility with an in-band feedback mechanism for updating modulation/coding sets which is frequently proprietary. While reverse engineering [27] is possible, communicating updates over the existing data connection to the spacecraft can be more robust to protocol changes in vendor hardware.

B. Delay/Disruption Tolerant Networking

The CE-1 framework incorporates several concepts from Delay/Disruption Tolerant Networking (DTN). DTN is an architecture and set of protocols developed for space communication with contributions from standards bodies including CCSDS [33] and the Internet Engineering Task Force (IETF) [34]. This approach for CE-1 enables interoperability with multiple commercial and government agencies and builds upon a substantial foundation of knowledge in space networks.

The Deep Impact Network Experiment (DINET) [35] is one of the only flight experiments to demonstrate contact graph routing (CGR) [36], a type of DTN-layer routing, in a space network. The networking experiment utilized a four node topology consisting of two surface assets, a relay node, and a terrestrial node. The Integrated Laser Communication Relay Demonstration (LCRD) Low Earth Orbit User Modem and Amplifier Terminal (ILLUMA-T) [37] utilized contact plans and aspects of CGR between the International Space Station (ISS), three ground stations, and a multi-hop network of ground nodes. These experiments demonstrated the use of contact plans for internetworking DTN nodes together, however they also helped to uncover some potential shortcomings for the current approaches to CGR.

Realistically, a space communication link (radio or optical), will only be scheduled with a single ground station at a time. This decision will be made by a centralized scheduler which will plan a single path from space to ground. This information can be used in a DTN contact plan; however, the routing decision has already been made at the time of scheduling. In general, space networks are sparse and will have a limited number of potential paths to a destination. In addition, the physical link must be brought online prior to DTN layer services. This will require a communication schedule to be known and agreed upon, with software that coordinates between the various levels of the protocol stack to configure the full system properly.

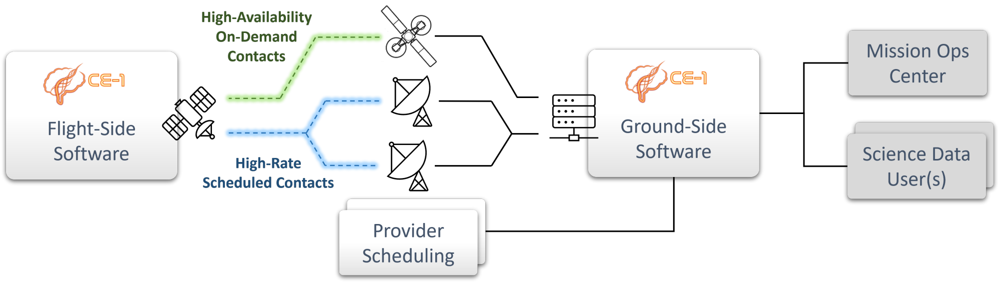


Fig. 1 Conceptual overview of CE-1. Software components exist on the spacecraft and ground, connected by RF links through provider ground stations and relay satellites. Ground software has connections for scheduling service with providers and transferring data to end users.

C. Service Acquisition

Prior work has conceptualized [6, 38] and experimentally demonstrated [39, 40] the concept of automated scheduling based on real-time needs of the user spacecraft in the near-Earth regime. Similar techniques have been proposed for cislunar [41] and deep space [42]. The Proximity-1 protocol [43] is used to establish a hailing channel over which Martian spacecraft can request service from an orbiter.

The problem of optimally scheduling ground stations within a provider's network based upon mission needs has received considerable study [44–47]. Less attention has been paid to *multi-provider* scheduling in which an agent has incomplete control over each of several networks. The key decisions in this environment are which provider and which configuration are used to meet a mission's needs. Multi-provider frameworks may require services to integrate a compatible interface into their systems [48, 49]. This common interface decreases complexity but may hinder adoption.

D. Evaluation of Space Communications Systems

Provider networks frequently operate compatibility test facilities to verify spacecraft communications before launch [50–52]. NASA's Electronic Systems Test Laboratory provides end-to-end pre-launch testing across several networks used throughout the mission lifetime of a crewed spacecraft [53]. These facilities operate at very high fidelity with replicas of operational hardware. However, they serve primarily to test against existing capabilities rather than research new ones.

Several satellite communications testbeds leverage commodity software-defined radios, primarily for evaluation of physical layer techniques [54, 55]. Spacecraft networking testbeds driven by orbital mechanics engines such as Systems Tool Kit (STK) have been used in DTN research [56, 57]. In [58], the authors present an STK-driven emulation environment of space-Earth communications primarily for interference mitigation research. Several recent testbeds have incorporated over-the-air connections through a relay satellites focusing on satellite-enhanced 5G communications [59, 60].

Parameter	Description
Destination	Final endpoint for data.
Priority	User-defined data priority. Values [0-2] correspond to bulk, normal, and expedited priority level.
Creation Time	Timestamp of bundle creation. Used in results section for latency metrics.
Deadline	Time after which payload will no longer be useful. Used by CE-1 as a target latency.

Table 1 Selected bundle header fields used by CE-1.

III. System Features

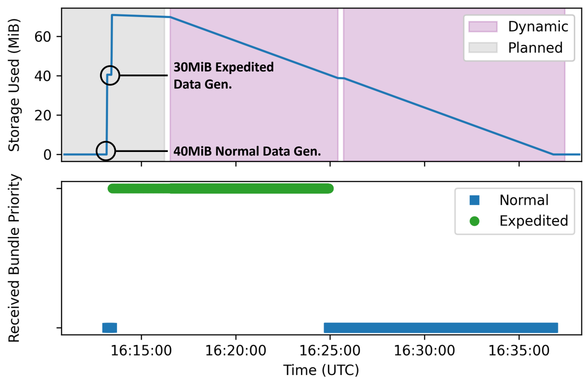
The proposed framework is a distributed system with components in the space and ground segments (Fig. 1). Wireless links through provider ground stations and relay satellites primarily transfer mission data but also exchange metadata between the space and ground components of CE-1. This section will discuss three key steps in the communications process which the system automates: monitoring mission data, scheduling service, and managing protocol compatibility during contacts. Where appropriate, illustrative examples are provided to highlight capabilities of each feature.

A. Automatic Data Handling

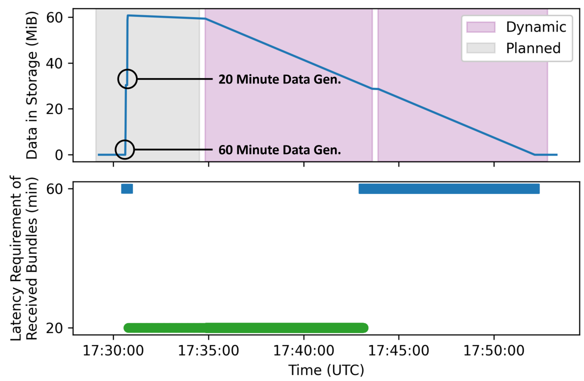
Transfer of mission data uses the bundle protocol [61], developed for reliable data transmission in networks with intermittent connectivity and large delays. Bundles are either created directly by spacecraft subsystems (vehicle health, science instruments, etc.) or converted from files or telemetry streams via bundling agents. Metadata contained in bundle headers (Table 1) is used by CE-1 to make decisions. Specifically, CE-1 will autonomously execute steps in the communications process to transfer data to its destination by the deadline. This includes determination of a next-hop, acknowledged transfer of bundles, and monitoring need for additional capacity.

Parameter	Description
Source	Node address of data sender.
Destination	Node address of data receiver.
Start	Beginning timestamp of contact availability.
End	End timestamp of contact availability.
Data Rate	Anticipated link capacity used for CGR calculations.
Radio ID	Identifier for radio which is compatible for the given configuration.
Priority	Used to configure primary/secondary routes to the same destination.
Config. ID	Mission-specific identifier of the provider and its configuration parameters (e.g. frequency, rate).

Table 2 Contact plan parameters used by CE-1. Standard CGR parameters are above. Additional parameters added for CE-1 use are below.



(a)



(b)

Fig. 2 Example of bundle offload by (a) priority and (b) deadline. In (a) expedited bundles take precedence over normal-priority bundles. In (b) bundles with normal priority are generated with two deadlines and the nearer deadline takes precedence.

1. Routing and Storage

CGR operates on a *contact plan*: an enumeration of all contacts available to the spacecraft along with their start and stop times (Table 2). Software onboard the spacecraft selects a next-hop node with positive advance towards the destination in the bundle header [62]. Route selection uses the earliest arrival time metric [63]. An end-to-end route is not required to be simultaneously available. Occasionally CGR will determine it is more optimal to select a future contact for bundles or, more simply, there may be periods of time when a next-hop is not available such as periods of no connectivity between passes to ground stations. In this case, bundles are buffered in onboard storage and released when the next-hop node is available.

2. Transfer

As will be seen in Section III.C, Internet Protocol (IP) is used throughout provider networks. Convergence layers carry bundles over intermediate hops in provider networks and the wider terrestrial Internet, which may or may not be DTN nodes. CE-1 uses a Licklider Transmission Protocol (LTP) convergence layer to carry spacecraft data over UDP/IP space-Earth links. Terrestrial data distribution (i.e., between CE-1 ground side software and mission operation centers) uses a TCP convergence layer over TCP/IP. During an active contact, LTP packets are transmitted at an initial rate set in the contact plan. Real-time updates to this rate limit may be applied for ACM links. Acknowledgements are provided by the next-hop node and allow the sender to release transferred data from storage.

Recurring LTP pings, sent from the spacecraft to the next hop during an active contact, are utilized alongside CGR for detecting unexpected link outages. If a ping is acknowledged, the route to the next hop is considered active. Otherwise, the route is excluded from route options. Whenever a route changes physical state, the preferred route is recomputed. This effectively enables a spacecraft to automatically switch to a secondary radio when the primary fails.

Data offload is sorted first by priority (bulk, normal, expedited). Within a class of data, bundles with the nearest deadline will be transferred first. An example is shown in Fig. 2a. At 16:13:10, normal priority data is generated. This begins flowing over the planned link. At 16:13:25, expedited priority data is generated and CE-1 begins transferring this data instead. Downlink of normal priority data begins after all expedited bundles are received. Likewise, in Fig. 2b bundles with a deadline 20 minutes in the future are prioritized over bundles with the same priority but a later deadline of 60 minutes.

3. Storage Monitoring

At regular intervals, CE-1 will evaluate bundles in storage against the capacity of the contact plan to determine if: 1) data will not be transferred by its deadline; and 2) onboard storage is at risk of exhaustion. If either condition exists, the spacecraft will generate a request for additional capacity and send it to ground-side software. Section IV.A describes this assessment in detail. As results will show, this enables automatic recovery after failed contacts. Issues with downlinks (e.g., interference, mechanical failure, provider configuration issues) occur occasionally and indeed are far more common than issues with the spacecraft itself [64]. After a failed contact, unacknowledged data will remain in onboard storage. CE-1's storage monitoring process will generate requests to schedule replacement contacts. These requests include the data volume requested, time required by, and a priority.

B. Automatic Scheduling

Broadly speaking, we can divide services into two categories representing the trade-off between between performance and availability. *On-demand* services require no pre-coordination before a link can be used by virtue of having a receiver always ready for a user's transmissions. These services frequently employ multiple access schemes which divide resources (i.e. in the time, frequency, or code domains) and place a limit on per-user capacity. Comparatively, *scheduled* services are often single-user assets and use schedules to determine which spacecraft is being tracked by a given beam or reflector antenna. These services are comparatively higher performance but have less availability. As an example, consider NASA's Tracking and Data Relay Satellite System (TDRSS) which provides both an on-demand [65] and scheduled service. Though both services are relayed through the same satellite the performance difference is large: 300kbps versus 1.5Gbps, respectively [66]. As the name implies, scheduled services must be reserved with the provider in advance.

Current practice for many space missions involves a human-driven forecast scheduling process to request contacts [67]. This requires operators to anticipate future data transfer needs days or weeks in advance which can lead to either over-scheduling of resources or failure to schedule enough capacity for actual data volumes. However, some providers offer the ability to schedule excess capacity through automated interfaces with little lead time. CE-1 uses this capability to acquire additional capacity when required for spacecraft data transfer. Since provider scheduling systems

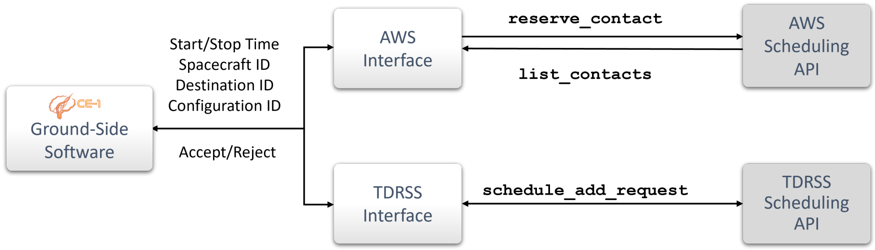


Fig. 3 Translation between generic contact format used by CE-1 components (white) and the scheduling API calls of two example service providers (gray). The provider's response is translated into CE-1's generic accept/reject format.

are frequently accessed via an application programming interface (API) over the Internet, ground-side CE-1 software performs this function on behalf of the spacecraft. This process involves receiving a request message, interfacing with one or more provider APIs to schedule service, and returning an acknowledgement to the spacecraft.

1. Contact Selection

On the ground, CE-1 generates a list of candidate line-of-sight contacts between the mission spacecraft and each provider asset. The optimal contact is selected following the algorithm described in Section IV.B. CE-1 communicates the request to the associated provider which will either confirm the request or reject it (frequently because the selected asset is occupied by another customer at the time). If the top-ranked contact is not available, CE-1 attempts to schedule the second-ranked and so on.

If desired, the software can batch process requests received from several spacecraft within a configurable window. Multiple requests are handled in priority order. The software has been stress tested with 51 emulated spacecraft generating 831 requests over a 72 hour period. With appropriate computing power, scaling to an arbitrary number of nodes is possible. CE-1 software manages contacts in generic data structures with a common set of fields for all providers. A set of modular interface layers translates between CE-1 and API calls to a specific provider's scheduling interface. Figure 3 shows an example of this translation for the scheduling systems of TDRSS and Amazon Web Services (AWS) Ground Station [68].

2. Request/Response Transfer

Service requests must reach ground software potentially in the absence of a scheduled contact. We assume that spacecraft have persistent access to an on-demand service, discussed below, which allows low-rate exchange of information without the need to schedule with a provider in advance. CE-1 flight software uses these channels to send requests to ground software and receive responses. In this way, on-demand contacts are used to provision scheduled contacts.

Several options for on-demand services exist. Past work has demonstrated scheduling requests sent through the

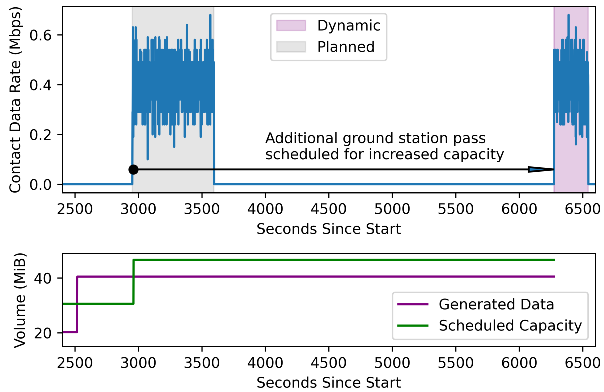


Fig. 4 In the absence of on-demand contacts, CE-1 can use existing scheduled contacts (gray) to send requests for additional contacts (purple).

TDRSS Demand Access System (DAS) [39]. Spatial and code diversity allows simultaneous reception of signals from several spacecraft. No equivalent on-demand service exists in the opposite direction, but a bidirectional link can be established with minimal lead time [7]. Current and upcoming commercial alternatives exist [69–71]. Another option demonstrated in [72] involves near-omni antennas colocated at ground stations to provide a low-rate control channel without the need to slew ground station antennas.

Additionally, requests can be sent in-band during existing scheduled contacts to set up future service [67]. Figure 4 shows a representative test. This scenario emulates a low-power CubeSat unable to close a link to relay satellites even at low rates. At the start, the spacecraft has a planned contact with sufficient capacity for its stored data. However, at approximately 7 minutes out, additional high-priority data is generated by spacecraft instruments. The request and response are sent in-band during the planned contact. This results in CE-1 scheduling an additional ground station pass to transfer the remaining data.

C. Automatic Service Execution

To send data through a service provider's network, the spacecraft must be compatible with protocols at the physical, data link, and network layers. CE-1 ensures these provider-specific protocols are correctly configured before each contact and adjusted in real-time as needed. Many service providers, particularly those whose services connect terrestrial users to the Internet, support IP throughout their networks from the air interface through data delivery to end users. CE-1 uses IP as a common layer for routing of metadata between space and ground-side software. Mission data (via bundle protocol convergence layers) is sent to the IP address of its destinations such as mission operations centers, science data users, etc. Intermediate nodes need not be DTN-enabled so long as they can route IP. In the uplink direction, data is sent to a common spacecraft IP. CE-1 ensures this data is routed through the proper provider as each contact is started and stopped.

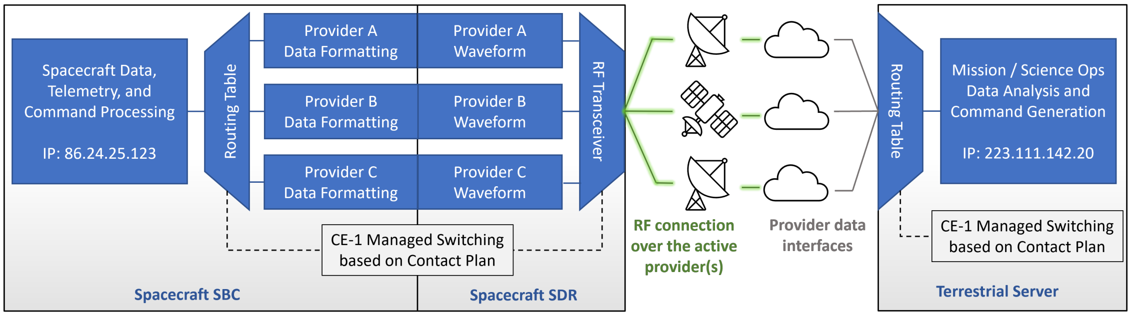


Fig. 5 Overview of CE-1's automatic service execution capability with pseudo-IP addresses. Compatible protocols are configured on the spacecraft SBC and SDR for each contact. The spacecraft and destination are reachable at the same persistent IP addresses even as the active provider changes.

1. Provider Protocol Sets

We assume CE-1 flight software runs on a Linux single-board computer (SBC) with interfaces to spacecraft radios. For each provider, CE-1 instantiates a Linux kernel network device (Fig. 5). Each interface is assigned an IP address which can reach a provider's IP gateway. The interface also implements required data framing for protocol compatibility [73]. Framed bytes are transferred to spacecraft radios over a local data interface (e.g., Ethernet, SpaceWire, etc.). Radios implement physical-layer protocols. While CE-1 can support provider-specific terminals, ideally this is a software-defined radio (SDR). A single SDR can be reconfigured with firmware for the active-contact's protocol set [26, 74].

2. Link Switching

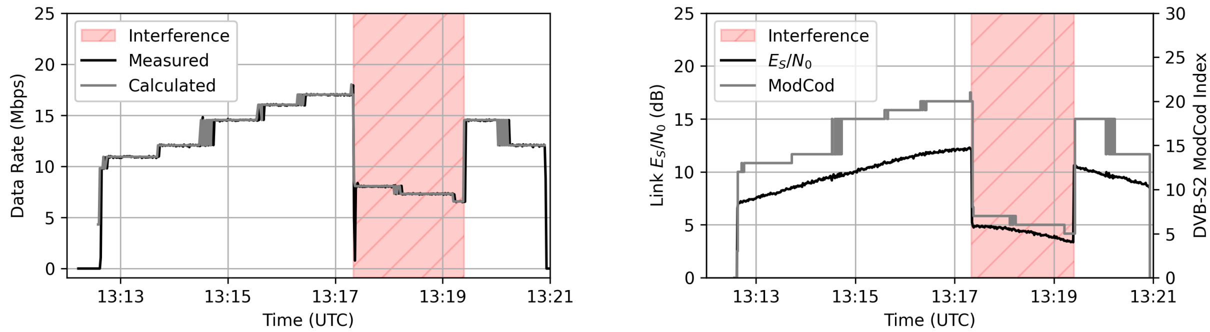
At the start of each contact, CE-1 uses information from the contact plan (Table 2) to:

1) configure the corresponding radio with proper physical-layer protocols and begin RF transmission.
2) adjust the Linux routing table so traffic flows over the corresponding network interface. This configuration also occurs on ground-software so traffic to the spacecraft is routed to the active provider's data interface.

This process allows the mission operations center to send IP traffic to the spacecraft's address (`86.24.25.123` in Fig. 5) with the knowledge it will reach the spacecraft as it roams between Provider A, B, and C. Simplification of operations via automation allows spacecraft to operate with more providers than would be possible if manual configuration of each contact by mission operators was required. This in turn enables a large set of candidate contacts which can easily be drawn upon to provide capacity to the spacecraft as needed.

3. Rate Adaptation

Standards which support ACM such as the ETSI Second Generation Digital Video Broadcasting – Satellite (DVB-S2) [29] have been used in spacecraft communications. However, DVB-S2 does not standardise a feedback mechanism for modulation/coding (ModCod) updates, leaving this as a vendor implementation detail. To improve compatibility



(a) (b)

Fig. 6 ModCod updates through the active link to flight-side software allows real-time adjustment of rates for interference and time-varying path loss.

across many service providers using DVB-S2, CE-1 instead provides ModCod updates as IP packets which flow from ground-side software to the spacecraft over the active link [75]. Onboard the spacecraft, ModCod updates are passed to the radio. Additionally, the network rate of transmitted bundles is adjusted to match the new link capacity. Figure 6 shows this adjustment will optimize capacity in the presence of time-varying link conditions and intermittent interference.

IV. System Decision Making

Built on top of the automation framework detailed in Section III, CE-1 makes two core decisions: when additional service is required and what contact to allocate in response to a user's request.

A. Capacity Assessment

Since CE-1 offloads data by priority and then deadline, it follows the same order when evaluating capacity. Priority $p \in \{0, 1, 2\}$ corresponding to bulk, normal, and expedited, respectively. We group deadlines into W windows of duration d for evaluation. The windows are specified in offsets from current time, $\{0 - t_d, t_d - t_{2d}, \dots, t_{(W-1)d} - t_{Wd}\}$. The process begins by summing the total volume of data in the wth window with priority p

$$V_{p,w} = \sum_i v_i \cdot \mathbf{1}_{\{p_i=p,\ t_{(w-1)d} \le t_i \le t_{wd}\}} \tag{1}$$

where v_i is the volume of the ith bundle and $\mathbf{1}_{\{\cdot\}}$ is the indicator function. The indicator function is 1 if the conditions concerning priority p_i and deadline t_i of the ith bundle are true, 0 otherwise.

There are N contacts in the spacecraft's contact plan with rate r_n over an interval of offsets relative to current time $[t_{0,n}, t_{1,n}]$. Capacity of the nth contact over a time window that begins in t_b and ends in t_e seconds is

$$c_n(t_b, t_e) = r_n \cdot \max\left[0, \min(t_e, t_{1,n}) - \max(t_b, t_{0,n})\right]. \tag{2}$$

Total capacity from now to the end of the wth window is $C(t_{wd}) = \sum_{n=1}^{N} c_n(0, t_{wd})$. If capacity is insufficient to offload data by the deadline we must request additional volume of data

$$\Delta_{p,t_{wd}} = \max\left(0, V_{p,w} - \left[C(t_{wd}) - \sum_{k<p}^{l<w} V_{k,l}\right]\right). \tag{3}$$

The bracketed term in (3) represents the remaining capacity after accounting for higher-priority requests and nearer deadlines. The set of requests $\mathcal{R} = \{\Delta_{p,t_{wd}} \mid p \in \{2, 1, 0\},\ w \in 1, \ldots, W\}$ is ordered first by priority and second by deadline.

If there are no requests required to meet deadlines ($\mathcal{R} = \emptyset$) the process then checks for storage exhaustion. Total storage usage is

$$V_S = \sum_i v_i' \tag{4}$$

where v_i' is the disk space occupied by the ith bundle. This differs from v_i due to the block size of the file system used for storage. For example, the Linux `ext4` file system aligns data in 4KiB blocks so $v_i' = 4096 \cdot \lceil v_i/4096 \rceil$. Due to this alignment, throughout this work we will specify data volumes in units of 1024 bytes using kibibytes (KiB), mebibytes (MiB), and gibibytes (GiB) rather than the more common base 1000 units of kB, MB, and GB. The mechanism will be triggered when total storage use exceeds a percentage threshold τ_{trig} of the total available disk space V_A. To reduce disk utilization to a target percentage τ_{target}, a high-priority request is generated

$$\Delta_{2,t_x} = \begin{cases} V_S - \tau_{\text{target}} V_A - C(t_x), & V_S/V_A > \tau_{\text{trig}} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where t_x is the time offset when storage issues must be resolved. A mission sets t_x approximately equal to time for storage to fill from threshold τ_{trig} to 100% under a worst-case data generation rate. The set of $\mathcal{R}$ requests will be sent to the ground. We will wait $|\mathcal{R}| \cdot t_{\text{rtt}} + t_{\text{loop}}$ for the next loop where t_{rtt} is the round trip time to fulfill one request and t_{loop} is a wait time parameter to allow enough data to gather before the next evaluation interval. This is summarized in Algorithm 1.

B. Service Fulfillment

Ground-side software receives the request message for volume Δ_{p,t_r} by time t_r with priority p. Orbit mechanics software generates list of possible contacts taking into account constraints. Primarily this is a line-of-sight calculation considering central-body obstruction, but can also include [72]:

1) Blockages onboard spacecraft.
2) Slew rate limits on gimbal or body-pointing motion.

Algorithm 1: Capacity assessment process.

while *True* **do**
    $\mathcal{R} \leftarrow \emptyset$;
    $t_{\text{wait}} \leftarrow t_{\text{loop}}$;
    for $p \in \{2, 1, 0\}$ **do**
        for $w \in \{1, \ldots, W\}$ **do**
            Calculate $V_{p,w}$ as in (1);
            $C(t_{wd}) \leftarrow \sum_{n=1}^{N} c_n(0, t_{wd})$;
            Calculate $\Delta_{p,t_{wd}}$ as in (3);
            if $\Delta_{p,t_{wd}} > 0$ **then**
                $\mathcal{R} \leftarrow \mathcal{R} \cup \{\Delta_{p,t_{wd}}\}$;
            end
        end
    end
    if $\mathcal{R} = \varnothing$ **then**
        $V_S \leftarrow \sum_i v_i'$;
        Calculate $\Delta_{2,\text{tx}}$ as in (5);
        if $\Delta_{2,tx} > 0$ **then**
            $\mathcal{R} \leftarrow \mathcal{R} \cup \{\Delta_{2,\text{tx}}\}$;
        end
    end
    foreach Δ_{p,t_r} *in* $\mathcal{R}$ **do**
        Send priority p request for Δ_{p,t_r} bytes by t_r;
        $t_{\text{wait}} \leftarrow t_{\text{wait}} + t_{\text{rtt}}$;
    end
    Sleep t_{wait};
end

3) Sun illumination if a spacecraft requires active solar to remain power positive.

4) Time constraints during which mission operators do not wish communications to occur.

If a provider supports multiple configurations (e.g., data rates, power levels), a candidate contact is created for each configuration. Each contact has an associated required symbolwise signal-to-noise ratio $E_S/N_{0\text{req},n}$ dictated by provider hardware and its configuration. A link budget is calculated

$$E_S/N_{0,n}(t) = \text{EIRP}_n + \text{G/T}_n - 20\log_{10} d_n(t) - 20\log_{10} f_{c,n} - 10\log_{10} R_{S,n} + 376.15 \tag{6}$$

where EIRP is the spacecraft's radiated power in dBW. Provider configuration-specific parameters G/T, f_c, and R_S are receive antenna gain-to-noise-temperature, center frequency, and symbol rate, respectively. The distance $d_n(t)$ in meters between spacecraft and provider asset is solved by orbit mechanics software over the contact interval $[t_{0,n}, t_{1,n}]$. The condition function

$$\Gamma(t) = E_S/N_{0,n}(t) > E_S/N_{0\text{req},n} + \text{margin} \tag{7}$$

indicates when the link closes with the desired margin (3dB is a typical value). A revised interval

$$
\begin{aligned}
t'_{0,n} &= \min\{t \mid \Gamma(t),\ \forall t \in [t_{0,n}, t_{1,n}]\} \\
t'_{1,n} &= \max\{t \mid \Gamma(t),\ \forall t \in [t_{0,n}, t_{1,n}]\}
\end{aligned} \tag{8}
$$

is calculated such that the link is possible over $[t'_{0,n}, t'_{1,n}]$. Contacts in which the link never closes are removed from consideration.

We calculate capacity from (2) using each candidate's adjusted times. If the contact is longer than needed, stop time is then trimmed so the contact only covers time required to transfer the requested volume plus a fixed duration t_{acq} to account for modem acquisition

$$
t''_{1,n} = \min\left\{t \mid t'_{0,n} + t_{\text{acq}} \le t \le t'_{1,n},\ c_n(t'_{0n}, t) \ge \Delta_{p,t_r}\right\}. \tag{9}
$$

Start time does not change, but we will use $t''_{0,n} = t'_{0,n}$ for notational simplicity and make the obvious notational substitutions for $t_{0,n}$ and $t_{1,n}$ when using (2) from this point on. Contacts which do not cover the minimum duration $t''_{1,n} - t''_{0,n} \ge t_{\min}$ are discarded. Finally, we also pare down contacts such that all occur before the request time t_r and none begin before a lead time t_{lead}. The lead time accounts for a worst-case time CE-1 takes to schedule service and get a response to the spacecraft. This results in a set of candidate contacts with capacities

$$
C = \{c_n(t_{\text{lead}}, t_r),\ \forall n\} \tag{10}
$$

For each contact, we define a utility function which maps mission preferences to the contact parameters. Each element of the equation has a fitness function $f \in [0, 1]$ and weight $\omega \in [0, 1]$ which indicates the relative importance the mission places on the corresponding quality. The utility function can contain many features such as quality of service, total volume, minimum duration, etc. Our studies show the most helpful features are volume, cost, and wait time

$$
U_n = \omega_{\text{cost}} f_{\text{cost,n}} + \omega_{\text{wait}} f_{\text{wait,n}} + \omega_{\text{vol}} f_{\text{vol,n}} \tag{11}
$$

The mission determines a relative cost $f_{\text{cost,n}} \in [0, 1]$ for it to schedule with the provider of the nth contact. We define

$$
f_{\text{wait,n}} = \max\left(\frac{t_{\text{wait}} - t_{0,n}}{t_{\text{wait}}}, 0\right) \tag{12}
$$

where t_{wait} is a scaling factor for the maximum desired wait time which depends on typical mission data. For example, a mission may set $t_{\text{wait}} = 3600$ if it is obligated to deliver science data no later than one hour after its collection. The volume fitness function is

$$f_{\text{vol,n}} = \max\left(\frac{c_n(t_{\text{lead}}, t_r)}{\Delta_{p,t_r}}, 1\right) \tag{13}$$

which equals 1 for candidates which will transfer all requested data. The system attempts to schedule $n^* = \text{argmax}_n(U_n)$ as described in Section III.B. If unsuccessful, the contact is removed from further consideration for this request and any future requests

$$C \leftarrow C \setminus \{c_{n^*}\}. \tag{14}$$

The process is run with the new set until success or no candidates remain. This is shown in Algorithm 2.

Algorithm 2: Service fulfillment process.

Solve line-of-sight contacts defined by $[t_{0,n}, t_{1,n}]$;
Trim contacts to $[t'_{0,n}, t'_{1,n}]$ as in (8);
Calculate capacities $C \leftarrow \{c_n(t_{\text{lead}}, t_r),\ \forall n\}$;
if $c_n(t_{lead}, t_r) > \Delta_{p,t_r}$ **then**
    Adjust end time to $t''_{1,n}$ as in (9);
end
Calculate U_n as in (11) for each candidate;
while $C \neq \varnothing$ **do**
    Attempt to schedule $n^* = \text{argmax}_n(U_n)$;
    if *success* **then**
        Send response to spacecraft;
        break;
    end
    $C \leftarrow C \setminus \{c_{n^*}\}$;
end

V. Testbed

The proposed development is evaluated on a testbed (Fig. 7) which provides a high-fidelity emulation of the space communications environment [13]. Realistic spacecraft and ground software environments, provider systems, radio hardware, and channels are used to test CE-1 capabilities in several scenarios.

A. Mission Spacecraft

The user spacecraft emulation focuses on the communication subsystem and flight computer data handling. Our chosen software-defined radio is a CesiumAstro SDR-1001, a smallsat form factor radio used in NASA's Starling mission [76]. A Linux computer hosts the space-side software components and connects to the SDRs over Ethernet. The testbed can be configured to have a single SDR support on-demand and scheduled contacts or to use a second SDR as a dedicated low-rate radio for on-demand contacts.

Testbed SDRs implement two protocol stacks ("waveforms") in the radio's programmable logic fabric for IP transfer

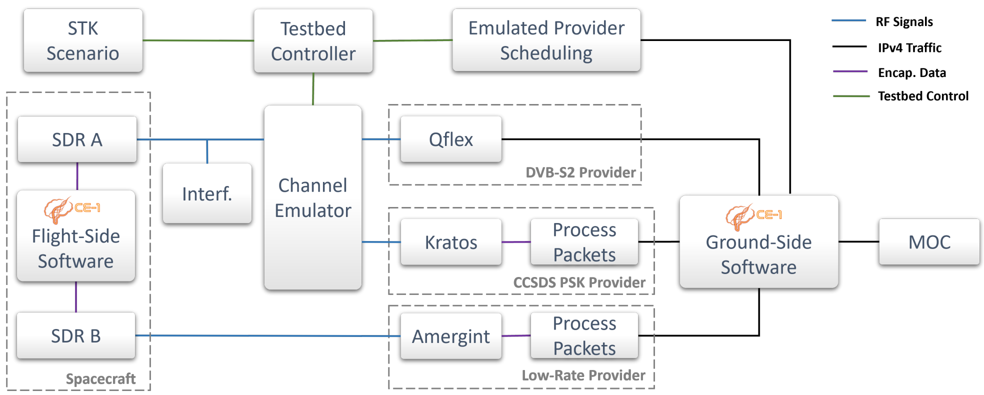


Fig. 7 Overview of components which comprise the testbed. SDRs are connected to provider modems through a channel emulator. Testbed controller software applies link parameters from STK to the channel emulator at contact start.

over wireless links: DVB-S2 Generic Stream Encapsulation (GSE) [77] and CCSDS Encapsulation Packet Protocol (EPP) [78]. Two CCSDS-compliant physical-layer protocols are used. A high-rate QPSK waveform is configured primarily for scheduled links. A spread-spectrum waveform compatible with TDRSS' multiple access (MA) service is used for a low-rate on-demand channel. Current achievable data rates over these links are shown in Table 3.

B. Provider Emulation

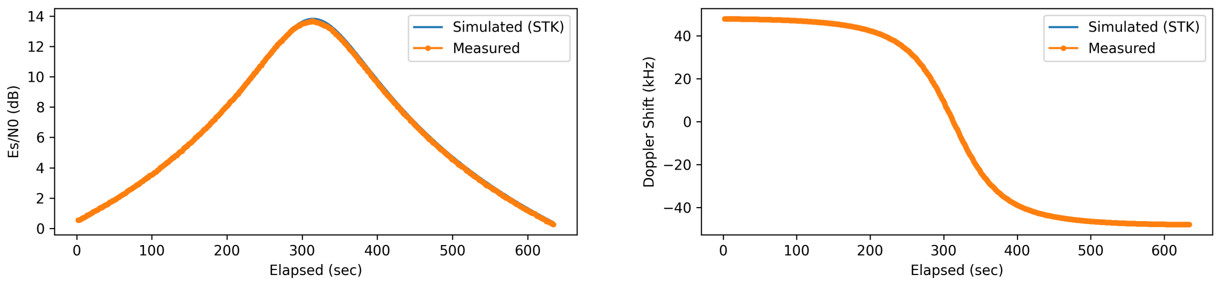
The testbed includes emulated provider systems for each link type in Table 3. Rackmount modems represent the hardware a service provider would install at their ground site. The modems either exchange IP datagrams directly or communicate through a software interface for translation to routable IP. These interfaces (Fig. 7) stand in for software a provider would instantiate at their ground site to handle IP traffic. To demonstrate CE-1's dynamic scheduling ability without committing antenna time, the testbed requires the ability to emulate responses from scheduling APIs of service providers. A generic emulated service provider responds to scheduling requests from CE-1. This software accepts or rejects requests with a configurable uniform probability.

C. Channel Emulation

Signals from spacecraft radios and provider modems are connected by a Keysight Propsim F64 Radio Channel Emulator. Software makes and breaks connections to the scheduled modem based on the results of CE-1 scheduling

Standard	Modulation	Encapsulation	Ground Modem	Max Rate
DVB-S2	PSK, APSK	GSE	Teledyne QFlex-400	15Mbps
CCSDS	QPSK	EPP	Kratos qRadio/qFEP	3.9Mbps
TDRSS MA	Spread-BPSK	EPP	Amergint sat-TRAC	193kbps

Table 3 Link types in testbed.



(a) (b)

Fig. 8 Simulated STK input versus measured testbed impairments show close agreement for an example contact. Specified parameters are (a) E_S/N_0 and (b) Doppler shift.

requests. Each testbed scenario is defined by a set of spacecraft, ground stations, and their RF parameters [79]. STK models the communication environment and orbital dynamics. Prior to a contact, testbed controller software queries STK for a profile of link parameters such as E_S/N_0, propagation delay, and Doppler shift. Time series of these impairments are loaded into the Propsim which applies them to the active link during a contact. Load time increases with emulation fidelity. We optimize accuracy for quick loading such that an example ground station pass (Fig. 8) is loaded in less than 10 seconds. Measurements show these settings produce emulations accurate for Doppler within 1ppm, E_S/N_0 within 0.4dB, and propagation delay within 1%. The entire channel emulation process runs without human input, enabling long-duration tests.

The channel emulation can also induce off-nominal impairments. Most simply, testbed software can disconnect signals from modems at a preset time. This can simulate any number of provider equipment issues which would break the link entirely. The output of a Stanford SG380 signal generator produces additive in-band interference at adjustable signal power levels as in Fig. 6. Both these impairments can be scheduled repeatedly to introduce failure scenarios into long-duration tests.

D. Ground Software

Ground-side software is hosted on general purpose computers. Local network connections among the testbed emulate data transfer over the terrestrial Internet. These interfaces include connections to emulated provider data interfaces and emulated provider APIs. Additionally, an emulated mission operations center (MOC) receives spacecraft data and sources commands for uplink.

VI. Proof-of-Concept Scenarios

A series of tests were performed to validate unique capabilities of CE-1 as summarized in Table 4. Each scenario is motivated by a real mission albeit with simplifications and adjusted parameters for testbed limitations. All scenarios use parameters set according to Table 5.

	A	B	C
Latency-Constrained Delivery	•		•
Storage Exhaustion Prevention		•	
Service Provider Roaming	•	•	•
Data Prioritization	•		•
Link Failure Handling	•		

Table 4 Major CE-1 features tested in each scenario.

A. Handle Link Failures

1. Inspiration

The Joint Polar Satellite System (JPSS) consists of polar-orbiting weather satellites which provide data for forecasting and climate science. The JPSS-1 satellite (renamed NOAA-20 after launch) has latency requirements for end-to-end delivery of data products ranging from 80 minutes to 25 hours [80]. The mission requires these latency be met 95% of the time on a 30-day basis.

The primary downlinks of science data are via Ka-band links to ground station sites in Fairbanks, Alaska; Svalbard, Norway; and the antarctic research stations Troll and McMurdo [81]. A back-up link consists of a Ka-band transmission through TDRSS at the same rate. Primary commanding is through TDRSS S-band service at low kbps rates. Use of sites at both poles limits typical latency to a half-orbital period of approximately 50 minutes in nominal cases. However, periodic disruptions occur due to downlink issues: incorrect scheduling, mechanical issues, planned maintenance, power outages, or failure to acquire a signal. While simple configuration errors will only impact a single pass, hours-long outages are not uncommon. In these cases, it is typical for the mission to wait until the following scheduled ground station pass to recover data.

For example, consider an issue on October 17, 2022 recorded in JPSS-1's outage notifications [82]. No data was received during the Svalbard contact. This data was instead recovered during the following pass to McMurdo 50 minutes later. At the time of the McMurdo contact, the earliest-collected data onboard was approximately 100 minutes old, failing to meet the latency constraint. We applied windows from a historical database of TDRSS unscheduled time and

Parameter	Value	Parameter	Value
ω_{vol}	1.0	t_{acq}	25s
ω_{cost}	0.5	t_{min}	60s
ω_{wait}	0.1	t_{lead}	90s
t_{loop}	300s	t_{wait}	60s

Table 5 Parameters used in all test scenarios.

Issue Time (UTC)	Issue Description	TDRSS Available
5-Nov-2021 00:50	Ground hardware issues.	TDRS-12
17-Oct-2022 10:55	Data not received.	TDRS-10
24-Apr-2023 10:12	Wrong antenna scheduled.	TDRS-10

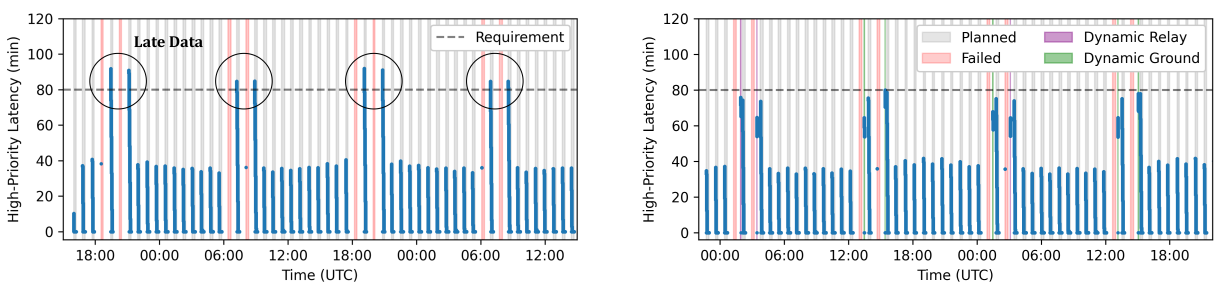
Table 6 Historical analysis of JPSS-1 data outages. In each case, a relay satellite was within line-of-sight and available for a duration of at least 20 minutes before the next pre-scheduled ground station pass.

found the TDRS-10 satellite was both within line-of-sight and available for scheduling within 15 minutes after the failed pass. With a typical 10 minute latency for scheduling TDRSS services [7], this contact could have been brought up to meet the latency constraint. Analysis of two other cases for which historical TDRSS unscheduled time was available (Table 6) also show backup contacts were available to meet the latency constraint.

There is substantial burden on the JPSS-1 operations team to manually track down the missing data products and uplink the necessary commands. It is therefore unlikely manual effort is worth it, since failures are infrequent enough that the 95% latency is still met on a 30-day average basis. However, CE-1 can handle these failures automatically without burdening operations personnel. The system can schedule additional service using backup ground stations and relays and re-transmit the data automatically to meet latency requirements.

2. Test Scenario

We develop a Mission A scenario, inspired by JPSS-1. The polar-orbiting Mission A satellite has access to three TDRSS relays which provide global coverage for low-rate commanding. The mission starts with planned contacts for every pass over Svalbard and McMurdo. These contacts have an average duration of 14 minutes and are separated by 50 minutes. In current configuration, the testbed cannot meet the actual rate used by JPSS-1 (though future software upgrades to emulated mission spacecraft will increase this limit). We set Mission A's data rate to 15Mbps and scale its data generation rate accordingly. We generate 2.0Mbps of lowest-priority (bulk) data which is given a deadline of 12



(a) (b)

Fig. 9 Latency between high-priority data generation and its delivery for (a) a baseline scenario with only pre-scheduled contacts and (b) an test scenario where CE-1 is enabled and contacts can be dynamically scheduled.

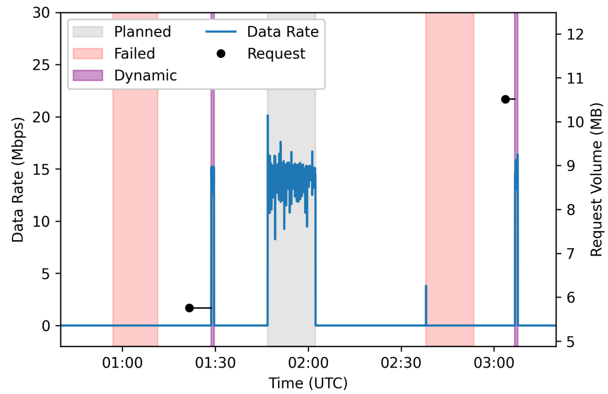


Fig. 10 Zoomed in view of contacts in the CE-1 scenario. Requests are generated after failed contacts in order to transfer data before its deadline.

hours from generation to downlink. Additionally, 1.2Mbps of high-priority (expedited) data is generated with a delivery requirement of 80 minutes. Under normal conditions this is 32GiB/day generated, a 30% margin over the 42GiB/day of scheduled capacity. Under nominal conditions, the planned contacts can transfer all generated data.

Testbed software simulates outages on back-to-back passes to McMurdo by disconnecting the RF signal from the modem. These outage pairs are repeated every 12 hours for the duration of a two-day test. We perform two test runs. The first is a baseline test without CE-1. In this run, data simply waits until the Svalbard pass and results in 50 minutes of additional latency after each outage.

In a comparison run, we enable CE-1 to meet the 80 minute latency for high-priority data. In addition to dynamic scheduling of TDRSS time, we give Mission A the ability to schedule service to ground stations. We emulate two generic ground station networks using locations of AWS and Kongsberg Satellite Services (KSAT) sites. Availability of all assets is set to 50% to emulate utilization by other customers which is beyond Mission A's control. We exercise the cost mechanism with $f_{\text{cost,AWS}} = f_{\text{cost,KSAT}} = 1.0$ (least expensive) and $f_{\text{cost,TDRSS}} = 0.1$ (more expensive). This will preference ground stations over relay satellite use. As will be seen, there will be gaps in ground station coverage, particularly in the Pacific, which will prevent ground station contacts before the latency constraint. In these cases the mission will schedule TDRSS despite increased costs.

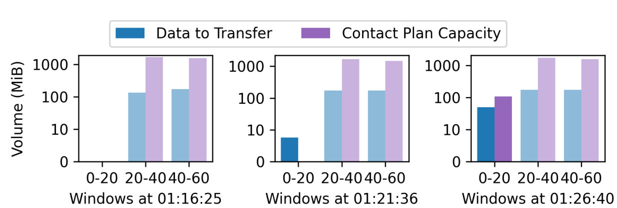


Fig. 11 Illustration of RDP logic around the 01:21 request. The highlighted 0–20 minute bin shows insufficient capacity until a request is generated.

Rank	Provider	Time	Utility
1	AWS HI	01:27:47 - 01:28:47	1.5957
2	KSAT HI	01:28:41 - 01:29:41	1.5950
3	AWS HI	01:28:47 - 01:29:47	1.5949
4	KSAT HI	01:29:41 - 01:30:41	1.5941
13	TDRS 08	01:24:43 - 01:25:43	1.1483
14	TDRS 11	01:24:43 - 01:25:43	1.1483
15	TDRS 08	01:26:24 - 01:27:24	1.1483
16	TDRS 11	01:26:24 - 01:27:24	1.1483

Table 7 Candidate contacts considered by CE-1 for the 01:21 request.

3. Results

With 12 hours for delivery and additional margin in the planned contacts, the low-priority data meets requirements in both runs. We focus on latency of high-priority data in the baseline case as shown in Fig. 9a. As expected, without manual intervention, failures drive the latency of the oldest data to nearly 92 minutes due to time until the next Svalbard contact. In comparison, in the run with CE-1 enabled, the request mechanism was triggered shortly after each failed contact. As in Fig. 9b, the dynamically-scheduled contacts ensure latency of high-priority data is no greater than 80 minutes.

Examining CE-1's response to the third pair of failed contacts is illustrative. Figure 10 shows these passes beginning with the first failed McMurdo pass around 01:00. Figure 11 visualizes the execution of Algorithm 1 onboard the spacecraft. At the 01:21, the first evaluation interval after the failed pass, CE-1 flight software notes the deadline of most bundles is greater than 20 minutes in the future. These will be fulfilled by the planned Svalbard contact beginning at 01:46. However, the oldest bundles were generated 60 minutes ago – immediately after the previous Svalbard pass ended. The deadline of these bundles, highlighted in Fig. 11, will be in less than 20 minutes for which there is no scheduled capacity. CE-1 flight software generates a request for this 6MiB of data with a required time within the next 20 minutes.

On Earth, CE-1 ground software generates a series of candidates as shown in Table 7. Since the requested 6MiB is small compared to the 15Mbps data rate, all contacts are trimmed to the minimum allowed duration of $t_{\text{min}} = 60$s. Though the TDRSS events occur sooner, they are ranked below the AWS contacts because $\omega_{\text{cost}} > \omega_{\text{wait}}$. CE-1 selects the first contact but this is rejected because the emulated AWS scheduling system has tasked AWS Hawaii to another customer at that time. Scheduling proceeds with the second ranked contact. This time the emulated KSAT scheduling system responds with a confirmation. The confirmation is passed to the spacecraft. This additional capacity is taken into account at the 01:26 evaluation interval and no additional requests are issued.

We encounter the same scenario after the second failed McMurdo pass in this group. The capacity evaluation occurs again at 03:03 and service is requested. Due to orbital procession, Mission A is now west of the Hawaiian Islands as

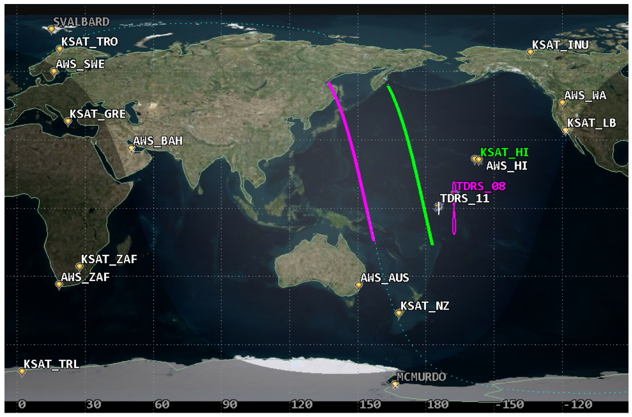


Fig. 12 Ground track of Mission A between request time and the 20 minute deadline for the 01:21 (green) and 03:03 (purple) requests. Schedulable ground stations (white) are shown along with the pre-scheduled McMurdo and Svalbard ground stations (gray) for reference.

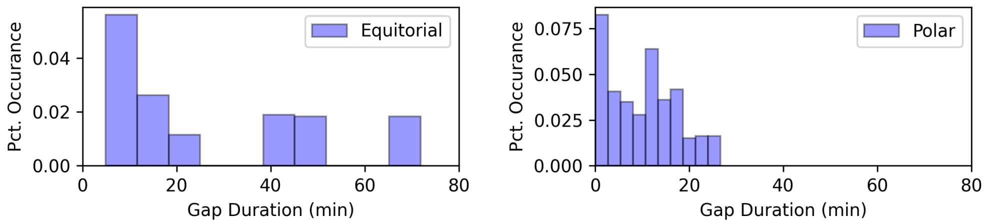
shown in Fig. 12. It only has TDRSS relays available so CE-1 schedules TDRS-08. Among the eight dynamic contacts scheduled in this test, two others went to TDRSS because they did not have ground station visibility. We note that without TDRSS we would not be successful in meeting 100% of latency over the run, but we would still improve over the baseline case for five of the eight failed contacts.

B. Handle Unexpected Data Generation

1. Inspiration

The Nuclear Spectroscopic Telescope Array (NuSTAR) is an X-ray telescope in Earth orbit capable of highly sensitive measurements of high-energy astrophysical sources. The NuSTAR communication system consists of an S-band transceiver with two omni-directional antennas providing nearly spherical coverage. NuSTAR primarily utilizes direct-to-Earth ground stations for commanding, telemetry, and science data, but can also communicate with TDRSS for low-rate commanding and telemetry.

Data is stored in one of four virtual recorders (ring buffers) on a solid-state recorder. Data generation rate of the photon detection instruments is proportional to the brightness of targets. Approximately 36-48 hours of observations at nominal brightness can be recorded before older data is overwritten. However, observations of the brightest targets will fill storage in just three hours and require frequent downlinks to prevent exhaustion. The mission plans passes in advance to meet the needs of its observation schedule. However, actual data generation rates can vary by a factor of two from estimates used during preplanning [64]. In these cases, science data is at risk of being overwritten before it can be downlinked. Conversely, if the actual generation rate is lower than expected, then the ground stations will be used inefficiently. The mission has invested heavily in automation for pre-planning [64], though these gains only bear fruit if predictions match reality. In current practice, mission operators will attempt to manually schedule additional service when the data rate is higher than expected to avoid storage exhaustion. CE-1's automated storage exhaustion mechanism



(a) (b)

Fig. 13 Histogram of gaps between ground station coverage for (a) NuSTAR's equatorial orbit and (b) the polar orbit chosen for Mission B. Simulation results over one week.

could handle these cases without requiring intervention of the mission operations team.

2. Test Scenario

We develop a Mission B scenario to test the storage exhaustion mechanism. Similar to NuSTAR, Mission B downlinks data over S-band at a rate of 2Mbps and has a low-rate S-band link to TDRSS for commanding. We also use NuSTAR ground stations at KSAT Singapore and the Italian Space Agency's ground station in Malindi, Kenya [83]. We pre-plan contacts against a nominal data generation rate of 45kbps and a storage volume of 312MiB. This results in four passes per day to each ground station. Under these nominal conditions, storage utilization does not exceed 75%. The baseline scenario will only use these pre-planned contacts, while the CE-1 test will also have the ability to dynamically schedule the same set of emulated AWS/KSAT sites as Mission A. Contact capacity and data deadline are set permissively such that the latency constraint mechanism is not triggered.

NuSTAR was launched into a nearly circular 630 km orbit with an inclination of 6°. While this equatorial orbit avoids impacts of the South Atlantic Anomaly, it does limit visibility to ground stations. Figure 13a shows a maximum gap of 70 minutes between candidate contacts. This can be an issue for dynamic scheduling since by the time CE-1 senses a problem, the requested contact will be too far in the future to make a difference. For this reason, Mission B is given a polar orbit similar to Mission A. This results in a maximum gap of 26 minutes (Fig. 13b).

3. Results

We again establish a baseline run without CE-1 enabled. A half day of the nominal scenario is shown in Fig. 14a which verifies our assumption that data will not exceed 75% of storage. In Fig. 14b, data generation rate is set to 90kbps. This represents a rate two times faster than predicted. Without manual intervention, onboard storage is exhausted. Next, the 90kbps scenario is rerun for a comparison test with CE-1 enabled. Approximately every 12 hours, data accumulates past the trigger threshold. CE-1 requests approximately 123MiB of data to return storage to target 40% of maximum. These requests are fulfilled by contacts to AWS and KSAT ground stations. As shown in Fig. 15, this prevents storage from being exhausted without manual intervention.

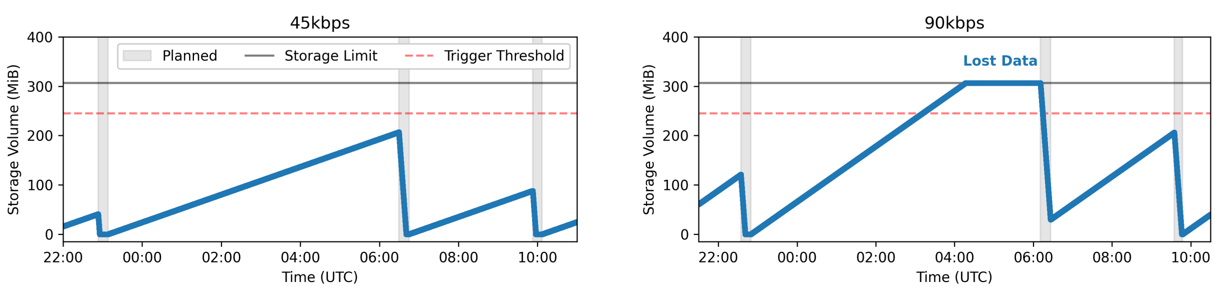
C. Reprioritize Data

1. Inspiration

NASA-Indian Space Research Organization (ISRO) Synthetic Aperature Radar (NISAR) mission is an observatory that will map the entire globe to determine changes in Earth's surfaces and ice masses. After its launch in 2025, NISAR will be supported by Ka-band links to NASA and ISRO ground station sites – including Fairbanks, Svalbard, and Punta Arenas in southern Chile. The mission plans for 15-20 contacts per day to NASA sites with 3.45Gbps data rates [84]. Radar data is stored in a high capacity solid state recorder until downlinked to the ground. Preserving storage space for later observations is important to the mission [85].

In addition to routine observations, NISAR will also be tasked with urgent requests for data over natural disaster sites [86]. The desired latency for these requests is less than five hours, though a significant portion of this is allocated to data processing on Earth. Fulfillment of these requests is on a best-effort basis and success will depend on the spacecraft's current storage capacity, planned contacts, and commanding ability. For example, data generated when storage is nearly full may require 9 hours for transfer unless the datasets are reprioritized. In this case, NISAR offloads data by priority and small data volumes can be downlinked within 2.5 hours with sufficient ground station availability.

In the future, government and commercial Ka-band relays may provide opportunity to offload these small, irregular observations. Though relay satellites won't fully replace the data volume of ground stations, their high availability could aid in the transfer of time-critical data. CE-1's ability to schedule service and automatically configure protocols would be of benefit for a mission interested in intermittent use of several relay service providers to meet tight delivery deadlines.



(a)

(b)

Fig. 14 Storage utilization in the baseline (without CE-1) case at nominal data rate (a) we do not exceed 75% of storage. At twice the predicted data rate (b), storage is exhausted and data is lost without manual intervention.

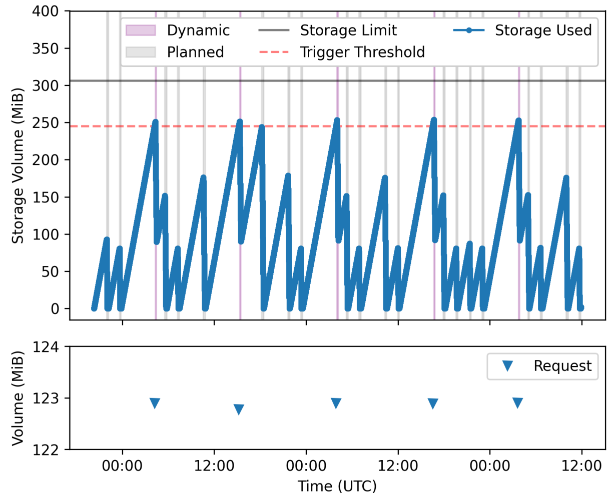


Fig. 15 Storage volume of Mission B over time. Exceeding the trigger threshold will generate requests for additional service. Over the two day run no data is lost due to storage exhaustion.

Fig. 16 Ground track of Mission C (blue) compared to planned ground stations (gray) and candidate relay satellites (white).

2. Test Scenario

In the Mission C scenario, approximately 15 contacts per day are planned across Svalbard, Puntas Arenas, and Fairbanks. The orbit is the same as Missions A and B. Since current testbed software cannot support NISAR data rates, we set Mission C's rate for these direct-to-Earth contacts at 15Mbps. At an average contact duration of 13.7 minutes, this equates to 21.6GiB of data per day. The spacecraft generates 1.6Mbps at a constant rate with a latency of 12 hours to emulate routine science observations. This results in 16.0GiB of data volume – a 35% margin over contact capacity. In addition, we randomly generate data to simulate urgent observations. These data are given expedited priority and a target latency of 30 minutes. Data arrival is modeled by a Poisson process with $\lambda = 2$ hours and volume is exponentially distributed with average of 50MiB.

Mission C also has the ability to transfer data over two emulated relay satellite constellations in geosynchronous orbit (GEO). In the test, we use orbital elements from TDRSS and the Inmarsat Global Xpress (GX) fleet. While each satellite constellation provides continuous coverage to LEO (Fig. 16), their availability is set to 50% in the testbed to emulate capacity used by other customers. In contrast to Mission A, we assume that the terminal is power-limited and links to GEO will be slower due to smaller receive antennas and increased path loss. Compared to the ground stations, we anticipate additional losses of roughly 10dB and 17dB to TDRSS and GX, respectively. Applying these scale factors to our direct-to-Earth data rate of 15Mbps, we arrive at 1.5Mbps and 0.25Mbps, respectively. Thus, while CE-1 can schedule GEO links to offload data it will take the data rate penalty into account in its decisions.

3. Results

In response to bursts of data, CE-1 will either use existing ground station passes or schedule a new relay contact. If a planned ground station pass is within 30 minutes of when the data is generated, CE-1 will not request additional service and instead automatically bump the high-priority data to the top of the queue. De-prioritized data will be offloaded after no high-priority data remains. With a 12 hour latency requirement and 35% scheduling margin on ground station contacts, it should not be an issue to transfer the lower-priority data on time. Figure 17 shows an example case from the CE-1 test run. Storage increases continuously due to the nominal data generated. At 03:34, an urgent observation is commanded which results in 72MiB of high-priority data. The capacity assessment sees that the capacity of the upcoming contact at 03:43 is sufficient. No request is issued, but the expedited data is offloaded first before resumption of the nominal data.

Alternatively, when no planned contacts exist within the 30 minute window, CE-1 will exercise its ability to dynamically schedule relay satellite service. No provider is favored over another from a cost perspective ($f_{\text{cost,TDRSS}} = f_{\text{cost,GX}}$). As before, CE-1 will schedule enough service to handle the high-priority data. Figure 18 shows these dynamic contacts scheduled to TDRSS and GX. We also note the significant variability in arrival time and data volume of randomly-generated data over the 2.5 day test.

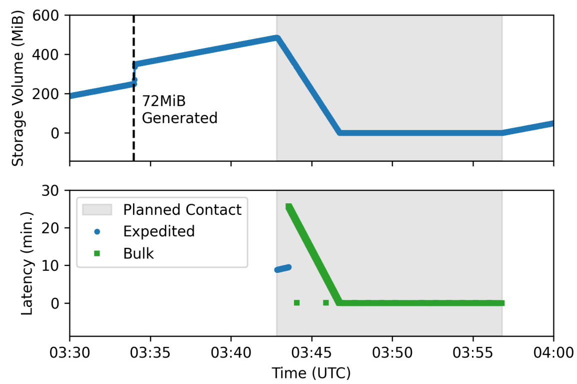


Fig. 17 Mission C's response when unplanned observations (dashed line) occur within 30 minutes of a planned contact. The high-priority data is offloaded first, followed by data from routine observations.

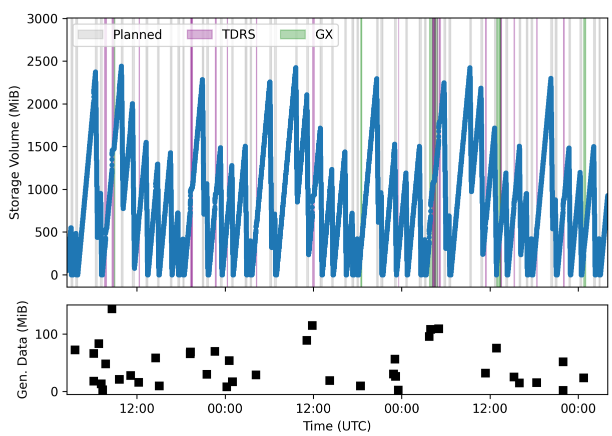


Fig. 18 Bursty data is randomly generated throughout the test to emulate unplanned observations. Data is offloaded from storage over planned ground station contacts or newly-scheduled relay satellite contacts.

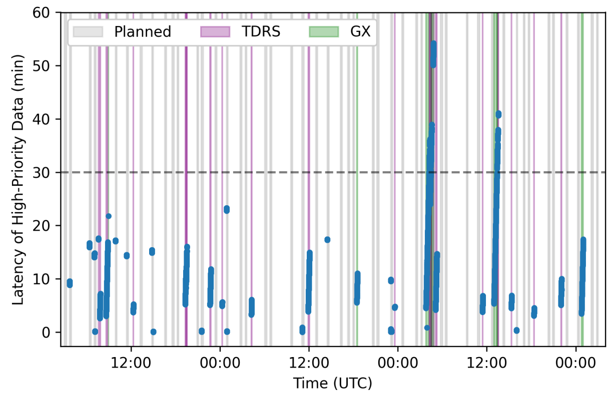


Fig. 19 Latency of high-priority data. With a few exceptions, data is transferred before the 30 minute deadline.

Figure 19 shows the resulting latency of the high-priority data. Across the two hour window, 93.2% of data by volume is delivered on time due to the two interventions of CE-1. Two periods around 04:00 and 13:00 are noted where data exceeds the 30 minute downlink deadline. In each case, all following conditions occurred:

1) No pre-planned contact was available within 30 minutes.
2) The request for higher-rate TDRSS link was rejected due to provider availability.
3) The generated data volume was greater than 53MiB, the capacity of a 30-minute GX contact at 0.25Mbps.

In these cases, meeting the latency was not physically possible. However, CE-1 makes repeated attempts to offload the data in a best-effort manner until it is transferred, scheduling additional service when available.

VII. Conclusion

A cognitive radio, as defined by the inventor of the term in [87], represents a progression from communication systems which are aware of their environment and able to adapt their parameters in response. Though an ideal cognitive radio also incorporates learning, the initial aware/adaptive systems still represent a significant improvement over state-of-the-art. A goal of CE-1 was to develop an autonomy framework upon which future cognitive engines (CE-2, CE-3, etc.) can incorporate intelligence. Replacing manual human processes with machine-to-machine communications provides a platform for integration of more complex machine learning algorithms. However, as these results demonstrate, this automated system is highly useful for space missions in and of itself.

We have shown how the framework continuously monitors mission data and communicates needs for additional service. The process of matching service requests with an optimal contact from a variety of service providers was detailed. The proposed concept has been developed into a suite of software deployed in a laboratory testbed which accurately emulates the space communications environment. Test results showed an ability to meet data latency requirements across different priorities, volumes, and deadlines. Additionally, storage exhaustion can be prevented when actual data generation rates do not match predictions. In each test case, the software automates the process of establishing end-to-end connections making transitions between providers seamless.

A shortcoming in the results of Section VI is that CE-1 will only react to data rather than proactively schedule based on data generation rates. This is most notable in Mission B, where the system will take no action on the increased data generation rate until onboard storage reaches the trigger threshold. After this threshold is reached, the system has a limited amount of time to schedule additional contacts necessitating a high priority for these requests. This was particularly pronounced in early tests with an equatorial orbit and long gaps between potential ground station contacts. Though storage exhaustion prevention functioned at a basic level in this orbit, the mechanism had to be triggered significantly earlier to prevent overflow in the time between the request and the dynamically-scheduled contact. A more reactive future system could observe data accumulating at twice the anticipated rate and proactively schedule additional service.

Future work will also investigate more complex routing, capacity assessment, advanced fault detection, and scheduling algorithms to build upon the framework presented here. Additionally, a subset of CE-1 software presented in this work – namely the automatic data handling and automatic scheduling features – were included in the TechEdSat-11 CubeSat which launched into low Earth orbit on a Firefly Alpha rocket in 2024. Results from on-orbit evaluation of these components is expected in early 2025.

VIII. Acknowledgements

We thank German Garcia and Amber Waid for their systems engineering expertise used to define CE-1 and its operations. Vladimir Linetsky and Taylor Pember were invaluable for construction of the testbed. Brian Frantz, Lucas Shalkhauser, and Peter Simon contributed waveform development required for testing. Stephen Paulus and Dean Schrage developed software to accomplish the automatic scheduling function. Ian Lim's clever scripting sped up configuration of test scenarios. Input from the HDTN team, particularly Brian Tomko, concerning the integration of the software into CE-1 was very welcome. Historical two-line element data from CelesTrak was of great use in the analyses presented here. NASA's Space Communication and Navigation Program, Advanced Communication and Navigation Technology Division, provided funding for development and demonstration of this work. The project was managed by Peter Schemmel until 2023 and Patrick Cleary afterwards.

References

[1] Baldwin, P. J., Evans, W. C., Guevara, D. L., Berner, J. B., Weir, E. L., and McCarthy, A. C., "NASA's Evolving Ka-Band Network Capabilities to Meet Mission Demand," *27th Ka and Broadband Communications Conference*, Stresa, Italy, 2022.

[2] Zavodsky, B., Dunion, J. P., Blackwell, W. J., Braun, S. A., Green, D. S., Velden, C., Adler, R. F., Cossuth, J., Murray, J. J., and Brennan, M. J., "Applications of NASA TROPICS Data for Tropical Cyclone Analysis, Nowcasting, and Impacts," *33rd Conference on Hurricanes and Tropical Meteorology*, Ponte Vedra, FL, 2018.

[3] Net, M. S., "Support of Latency-Sensitive Space Exploration Applications in Future Space Communications Systems," Ph.D. thesis, Massachusetts Institute of Technology, 2017.

[4] Baker, D. N., Riesberg, L., Pankratz, C. K., Panneton, R. S., Giles, B. L., Wilder, F. D., and Ergun, R. E., "Magnetospheric Multiscale Instrument Suite Operations and Data System," *Space Science Reviews*, Vol. 199, No. 1, 2016, pp. 545–575.

[5] Gehrels, N., Chincarini, G., Giommi, P., Mason, K. O., Nousek, J. A., Wells, A. A., White, N. E., Barthelmy, S. D., Burrows, D. N., Cominsky, L. R., Hurley, K. C., Marshall, F. E., Mészáros, P., Roming, P. W. A., Angelini, L., Barbier, L. M., Belloni, T., Campana, S., Caraveo, P. A., Chester, M. M., Citterio, O., Cline, T. L., Cropper, M. S., Cummings, J. R., Dean, A. J., Feigelson, E. D., Fenimore, E. E., Frail, D. A., Fruchter, A. S., Garmire, G. P., Gendreau, K., Ghisellini, G., Greiner, J., Hill, J. E., Hunsberger, S. D., Krimm, H. A., Kulkarni, S. R., Kumar, P., Lebrun, F., Lloyd-Ronning, N. M., Markwardt, C. B., Mattson, B. J., Mushotzky, R. F., Norris, J. P., Osborne, J., Paczynski, B., Palmer, D. M., Park, H.-S., Parsons, A. M., Paul, J., Rees, M. J., Reynolds, C. S., Rhoads, J. E., Sasseen, T. P., Schaefer, B. E., Short, A. T., Smale, A. P., Smith, I. A., Stella, L., Tagliaferri, G., Takahashi, T., Tashiro, M., Townsley, L. K., Tueller, J., Turner, M. J. L., Vietri, M., Voges, W., Ward, M. J., Willingale, R., Zerbi, F. M., and Zhang, W. W., "The Swift Gamma-Ray Burst Mission," *The Astrophysical Journal*, Vol. 611, No. 2, 2004, pp. 1005–1020.

[6] Reinhart, R. C., Schier, J. S., Israel, D. J., Tai, W., Liebrecht, P. E., and Townes, S. A., "Enabling Future Science and Human Exploration with NASA's Next Generation Near Earth and Deep Space Communications and Navigation Architecture," *68th International Astronautical Congress*, Adelaide, Australia, 2017.

[7] Roberts, C. J., Burke, J. C., Benson, M. J., Lubelczyk, J. T., Bradley, T. H., Heckler, G. W., and Hudiburg, J. J., "Evaluation of Timely Communications Access Methods Using NASA Space Network," *Journal of Aerospace Information Systems*, Vol. 18, No. 6, 2021, pp. 333–346. https://doi.org/10.2514/1.I010897.

[8] Seablom, M., Tan, F., Norton, C., Moses, J. D., and Kepko, L., "Advancing Technology for NASA Science with Small Spacecraft," *32nd Annual AIAA/USU Conference on Small Satellites*, Logan, UT, 2018.

[9] Hughes, K., di Pasquale, P., Babuscia, A., and Fesq, L., "On-demand Command and Control of ASTERIA with Cloud-based Ground Station Services," *2021 IEEE Aerospace Conference*, Big Sky, MT, 2021.

[10] Eilertsen, B., "NewSpace – forcing a rethink of ground networks," *14th International Conference on Space Operations (SpaceOps)*, Daejeon, Korea, 2016.

[11] Haddad, G., Kacpura, T., Nessel, J., Paulsen, P., Drexler, J., Hartman, E., Hayes, J., and Boldosser, P., "Framework for NASA Space Relay Satellite Service over the Next Decade: Development and Demonstration of the Commercial Successors to TDRS," *27th Ka and Broadband Communications Conference*, Stresa, Italy, 2022.

[12] Wong, Y., Schaire, S., Patel, C., Ambrose, L., Kegege, O., and Thorsen, D., "NASA Leveraging Commercial Communication Ground Stations for Small Satellites," *36th Annual Small Satellite Conference (SmallSat)*, Logan, UT, 2022.

[13] Downey, J., Gannon, A., Smith, A., Koch, M., Dudukovich, R., and Schweinsberg, E., "Emulated Spacecraft Communication Testbed for Evaluating Cognitive Networking Technology," *Cognitive Communications for Aerospace Applications Workshop*, Virtual, 2023.

[14] Chelmins, D., Briones, J., Downey, J., Clark, G., and Gannon, A., "Cognitive Communications for NASA Space Systems," *Advances in Communications Satellite Systems. Proceedings of the 37th International Communications Satellite Systems Conference (ICSSC)*, 2019.

[15] Neely, M., Modiano, E., and Rohrs, C., "Dynamic Power Allocation and Routing for Time-Varying Wireless Networks," *IEEE Journal on Selected Areas in Communications*, Vol. 23, No. 1, 2005, pp. 89–103. https://doi.org/10.1109/JSAC.2004.837349.

[16] Song, G., and Li, Y., "Cross-Layer Optimization for OFDM Wireless Networks – Part I: Theoretical Framework," *IEEE Transactions on Wireless Communications*, Vol. 4, No. 2, 2005, pp. 614–624.

[17] Liu, Q., Zhou, S., and Giannakis, G., "Queuing With Adaptive Modulation and Coding Over Wireless Links: Cross-Layer Analysis and Design," *IEEE Transactions on Wireless Communications*, Vol. 4, No. 3, 2005, pp. 1142–1153.

[18] Luo, C., Yu, F. R., Ji, H., and Leung, V. C. M., "Cross-Layer Design for TCP Performance Improvement in Cognitive Radio Networks," *IEEE Transactions on Vehicular Technology*, Vol. 59, No. 5, 2010, pp. 2485–2495.

[19] Ding, L., Gao, K., Melodia, T., Batalama, S. N., Pados, D. A., and Matyjas, J. D., "All-Spectrum Cognitive Networking through Joint Distributed Channelization and Routing," *IEEE Transactions on Wireless Communications*, Vol. 12, No. 11, 2013, pp. 5394–5405. https://doi.org/10.1109/TWC.2013.100213.120545.

[20] She, C., Yang, C., and Quek, T. Q. S., "Cross-Layer Optimization for Ultra-Reliable and Low-Latency Radio Access Networks," *IEEE Transactions on Wireless Communications*, Vol. 17, No. 1, 2018, pp. 127–141. https://doi.org/10.1109/TWC.2017.2762684.

[21] Hosseinian, M., Choi, J. P., Chang, S.-H., and Lee, J., "Review of 5G NTN Standards Development and Technical Challenges for Satellite Integration With the 5G Network," *IEEE Aerospace and Electronic Systems Magazine*, Vol. 36, No. 8, 2021, pp. 22–31. https://doi.org/10.1109/MAES.2021.3072690.

[22] Somerlock, O., Sharma, A., and Heckler, G. W., "Adapting Commercial 5G Terrestrial Networks for Space," *IEEE Aerospace Conference*, Big Sky, MT, 2022. https://doi.org/10.1109/AERO53065.2022.9843534.

[23] Vanderpoorten, J., and Zhang, K., "Flexible Modem Interface — Enabling DoD Wideband SATCOM Enterprise," *IEEE Military Communications Conference (MILCOM)*, Baltimore, MD, 2017, pp. 128–134. https://doi.org/10.1109/MILCOM.2017.8170723.

[24] Zeleznikar, D., "Flexible Modem Interface (FMI) in Space - Extending Standardized Commercial Satellite Communications Services to Space Users," *Ka and Broadband Communications Conference*, Sorrento, Italy, 2019.

[25] Heckler, G., Younes, B., Mitchell, J., et al., "NASA's Wideband Multilingual Terminal Efforts as a Key Building Block for a Future Interoperable Communications Architecture," *26th Ka and Broadband Communications Conference*, Arlington, VA, 2021.

[26] Piasecki, M., Downey, J., Pham, N., et al., "Development and Demonstration of a Wideband RF User Terminal for Roaming between Ka-band Relay Satellite Networks," *38th International Communications Satellite Systems Conference (ICSSC)*, Arlington, VA, 2021.

[27] Angert, M. P., Bubnash, B. M., Cerabona, M. A., Dauberman, M. G., Monaghan, C., Heistand, C. C., and Wilkes, J. C., "Frontier Radio – Multi-Lingual – A Next Generation Space Software Defined Radio," *IEEE Aerospace Conference*, Big Sky, MT, 2023. https://doi.org/10.1109/AERO55745.2023.10115782.

[28] Heckler, G. W., Piasecki, M. T., Narthelme, N. F., Schemmel, P. J., Weir, E. L., McCarthy, A. C., and Dobereiner, J. C., "NASA's Efforts to Pursue Commercial Communications Services for Missions in Near Space," *28th Ka and Broadband Communications Conference*, Bradford, UK, 2023.

[29] European Telecommunications Standards Institute, "Digital Video Broadcasting (DVB); Second Generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications (DVB-S2)," Standard EN 302 307 v1.2.1, Aug. 2009.

[30] Consultative Committee for Space Data Systems, "Flexible Advanced Coding and Modulation Scheme for High Rate Telemetry Applications," Standard 131.2-B-2, Feb. 2023.

[31] Downey, J., Mortensen, D., Evans, M., Briones, J., and Tollis, N., "Adaptive Coding and Modulation Experiment With NASA's Space Communication and Navigation Testbed," *International Communications Satellite Systems Conference (ICSSC)*, Cleveland, OH, 2016.

[32] Devaraj, K., Ligona, M., Blossom, E., Breu, J., Klofas, B., Colton, K., and Kingsbury, R., "Planet High Speed Radio: Crossing Gbps from a 3U Cubesat," *33rd Annual Small Satellite Conference (SmallSat)*, Logan, UT, 2019.

[33] Consultative Committee for Space Data Systems, "Delay Tolerant Networking Working Group Charter," cwe.ccsds.org/fm/Lists/Charters/DispForm.aspx, retrieved 21 August 2024.

[34] Internet Engineering Task Force, "Delay/Disruption Tolerant Networking Documents," , 2023. URL https://datatracker.ietf.org/wg/dtn/documents/.

[35] Wyatt, J., Burleigh, S., Jones, R., Torgerson, L., and Wissler, S., "Disruption Tolerant Networking Flight Validation Experiment on NASA's EPOXI Mission," *First International Conference on Advances in Satellite and Space Communications*, Colmar, France, 2009, pp. 187–196. https://doi.org/10.1109/SPACOMM.2009.39.

[36] Burleigh, S., "Contact Graph Routing," datatracker.ietf.org/doc/html/draft-burleigh-dtnrg-cgr-00, Internet Draft, retrieved 21 August 2024.

[37] Dudukovich, R., Raible, D., Tomko, B., Kortas, N., Schweinsberg, E., Basciano, T., Pohlchuck, W., Deaton, J., and Nowakowski, J., "Advances in High-rate Delay Tolerant Networking On-board the International Space Station," *IEEE Space Mission Challenges for Information Technology - IEEE Space Computing Conference (IEEE SMC-IT/SCC)*, Mountain View, CA, 2024.

[38] Israel, D. J., Heckler, G. W., and Menrad, R. J., "Space Mobile Network: A Near Earth Communications and Navigation Architecture," *IEEE Aerospace Conference*, Big Sky, Montana, 2016.

[39] Mortensen, D. J., Roberts, C., and Reinhart, R., "Automated Spacecraft Communications Service Demonstration Using NASA's SCaN Testbed," *15th International Conference on Space Operations (SpaceOps)*, Marseille, France, 2018.

[40] Gannon, A., Paulus, S., Gemelas, C., and Vincent, L., "Spacecraft-Initiated Scheduling of Commercial Communications Services," *39th International Communications Satellite Systems Conference (ICSSC)*, Stresa, Italy, 2022.

[41] NASA, "LunaNet Interoperability Specification Document," LN-IS, v4, Sep. 2022.

[42] Hackett, T. M., Johnston, M., Johnston, M., Wyatt, J., and Bilén, S. G., "Demand Access for NASA's Deep Space Network: A New Paradigm for Operations," *16th International Conference on Space Operations (SpaceOps)*, Virtual, 2021.

[43] Barbieri, A., Butman, S., Danos, M., Greenberg, E., Ilott, P., Kazz, G., Torgerson, J., Vaisnys, A., Adams, W., Johnson, C., Dapore, M., and Merz, D., "Development and flight performance of CCSDS Proximity-1 on Odyssey and the Mars exploration rovers," *IEEE Aerospace Conference*, 2005, pp. 1444–1454. https://doi.org/10.1109/AERO.2005.1559435.

[44] Barbulescu, L., Howe, A., and Whitley, D., "AFSCN Scheduling: How the Problem and Solution Have Evolved," *Mathematical and Computer Modelling*, Vol. 43, No. 9, 2006, pp. 1023–1037. https://doi.org/https://doi.org/10.1016/j.mcm.2005.12.004.

[45] Petelin, G., Antoniou, M., and Papa, G., "Multi-Objective Approaches to Ground Station Scheduling for Optimization of Communication With Satellites," *Optim Eng*, 2023, p. 147–184. https://doi.org/https://doi.org/10.1007/s11081-021-09617-z.

[46] Polich, K., Bode, B., and Reeves, A., "An Optimization Framework for Resource Allocation in Multi-Tenant Communication Networks," *IEEE Cognitive Communications for Aerospace Applications Workshop (CCAAW)*, Virtual, 2021.

[47] Spangelo, S., Cutler, J., Gilson, K., and Cohn, A., "Optimization-Based Scheduling for the Single-Satellite, Multi-Ground Station Communication Problem," *Computers & Operations Research*, Vol. 57, 2015, pp. 1–16. https://doi.org/https://doi.org/10.1016/j.cor.2014.11.004.

[48] Aalyria, "Spacetime: Orchestrating the Next Generation of Aerospace Networks," Whitepaper, 2024.

[49] Hadjitheodosiou, M., and Pirio, S., "A Pooling & Sharing Solution for Delivering Space Data Using Multiple Commercial Satcom Providers," *39th International Communications Satellite Systems Conference (ICSSC)*, 2022, pp. 263–267. https://doi.org/10.1049/icp.2023.1397.

[50] Schaire, S., Lamberson, M., Bundick, S., Roberts, C., Ambrose, L., Mason, J., Altunc, S., Wong, Y. F., Baros, J., Celeste, P., and Bollard, M., "Streamlining Ground Station Network Compatibility Test for Small Satellites," *15th International Conference on Space Operations (SpaceOps)*, Marseille, France, 2018. https://doi.org/10.2514/6.2018-2732.

[51] California Institute of Technology, "DSN Telecommunications Link Design Handbook: Test Support," 810-005, Rev. E, Oct. 2009.

[52] Flentge, M., Soerensen, E., Khan, Z., and Lugert, M., "Integration & Validation Approach in ESTRACK," *International Conference on Space Operations (SpaceOps)*, Huntsville, AL, 2010.

[53] NASA Johnson Space Center, "Electronic Systems Test Laboratory (ESTL) User Test Planning Guide," JSC-CN-24891, Jan. 2011.

[54] Duncan, J., Querol, J., Maturo, N., Krivochiza, J., Spano, D., Saba, N., Marrero, L., Chatzinotas, S., and Ottersten, B., "Hardware Precoding Demonstration in Multibeam UHTS Communications Under Realistic Payload Characteristics," *37th International Communications Satellite Systems Conference (ICSSC)*, Okinawa, Japan, 2019.

[55] Bui, H. C., and Franck, L., "Cost-Effective Emulation of Geostationary Satellite Channels by Means of Software-Defined Radio," *IEEE International Workshop on Metrology for Aerospace (MetroAeroSpace)*, Benevento, Italy, 2014, pp. 538–542.

[56] Komnios, I., Alexiadis, I., Bezirgiannidis, N., Diamantopoulos, S., Lenas, S.-A., Papastergiou, G., and Tsaoussidis, V., "SPICE Testbed: A DTN Testbed for Satellite and Space Communications," *9th International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TRIDENTCOM)*, Guangzhou, China, 2014. https://doi.org/10.1007/978-3-319-13326-3_20.

[57] Bull, R. L., Dudukovich, R. M., Fraire, J. A., Kortas, N., Kassouf-Short, R., Smith, A., and Schweinsberg, E., "Network Emulation Testbed Capabilities for Prototyping Space DTN Software and Protocols," *11th International Workshop on Computer and Networking Experimental Research using Testbeds (CNERT)*, Vancouver, Canada, 2024.

[58] Clark, S., Dorny, J., Law, S., Loving, J., and Werremeyer, M., "Space-Ground Communications Testbed," *34th Space Symposium*, Colorado Springs, CO, 2018.

[59] Evans, B., Wang, N., Rahulan, Y., Kumar, S., Cahill, J., Kavanagh, M., Watts, S., Chau, D.-K., Begassat, Y., Brunel, A.-P., Masson, T., and Diarra, M., "An Integrated Satellite–Terrestrial 5G Network and Its Use to Demonstrate 5G Use Cases," *International Journal of Satellite Communications and Networking*, Vol. 39, No. 4, 2021, pp. 358–379. https://doi.org/https://doi.org/10.1002/sat.1393.

[60] Corici, M., Liolis, K., Politis, C., Geurtz, A., Cahill, J., Bunyan, S., Schlichter, T., Völk, F., and Kapovits, A., "Satellite is 5G (SATis5) Whitepaper," Whitepaper, ESA, 2020.

[61] IET, "Bundle Protocol Specification," RFC5050, Nov. 2007.

[62] Mortensen, D., Eddy, W., Reinhart, R., and Lassere, F., "CNES-NASA Disruption-Tolerant Networking (DTN) Interoperability," *ISS Research and Development Conference*, Chicago, IL, 2014.

[63] Bezirgiannidis, N., Burleigh, S., and Tsaoussidis, V., "Delivery Time Estimation for Space Bundles," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 49, No. 3, 2013, pp. 1897–1910. https://doi.org/10.1109/TAES.2013.6558026.

[64] Roberts, B. A., Thorsness, J., and Bester, M., "Fully Autonomous Data Recovery with the NuSTAR Ground System," *AIAA SPACE 2013 Conference and Exposition*, San Diego, CA, 2013.

[65] Gatlin, T. A., Kearns, W., and Horne, W., "The NASA Space Network Demand Access System (DAS)," *International Conference on Space Operations (SpaceOps)*, Houston, TX, 2002.

[66] NASA Goddard Space Flight Center, "Space Network User's Guide," 450-SNUG Rev. 10, 2012.

[67] Israel, D. J., Roberts, C., Morgenstern, R. M., Gao, J., and Tai, W. S., "Space Mobile Network Concepts for Missions Beyond Low Earth Orbit," *15th International Conference on Space Operations (SpaceOps)*, Marseille, France, 2018.

[68] Amazon, "AWS Ground Station: API Reference Version 2019-05-23," docs.aws.amazon.com/ground-station/latest/APIReference, retrieved 21 August 2024.

[69] Iridium, "Short Burst Data," iridium.com/services/iridium-sbd/, retrieved 21 August 2024.

[70] Pang, T. K., and Trachtman, E., "Inter-Satellite Data Relay System (IDRS) for LEO Satellites Using a Commercially Available GEO Satellite System," *34th Annual Small Satellite Conference (SmallSat)*, Logan, UT, 2020.

[71] Lewis, B., "New Frontiers: NASA Partnership Validates Decades of Inmarsat Government's Customer-Focused Service and Innovation," inmarsatgov.com/new-frontiers-nasa, 2022.

[72] Gannon, A., Gemelas, C., Paulus, S., et al., "An Augmented Ground Station Architecture for Spacecraft-Initiated Communication Service Requests," *16th International Conference on Space Operations (SpaceOps)*, Cape Town, South Africa, 2021.

[73] Gannon, A., Brooks, D., Zogheib, S., Schweinsberg, E., Shalkhauser, L., Frantz, B., Koch, M., Dudukovich, R., Downey, J., and Briones, J., "Automated End-to-End Spacecraft Connectivity Across Diverse Links," *41st International Communications Satellite Systems Conference (ICSSC)*, Seattle, WA, 2024.

[74] Mortensen, D., Reinhart, R., Johnson, S., Briones, J., and Chelmins, D., "The Space Communications and Navigation Testbed aboard International Space Station: Seven Years of Space-based Reconfigurable Software Defined Communications, Navigation, and Networking," *International Space Station R&D Conference*, Atlanta, GA, 2019.

[75] Hilts, M., Gannon, A., Zogheib, S., and Schweinsberg, E., "Service Provider Agnostic Adaptive Coding and Modulation System Design," *41st International Communications Satellite Systems Conference (ICSSC)*, Seattle, WA, 2024.

[76] Grashuis, R., and Cannon, J., "Starling Swarm Mission: Technology Objectives, Status, and Future Applications," *SpaceOps 2024 Workshop*, Toulouse, France, 2024.

[77] ETSI, "Digital Video Broadcasting (DVB); Generic Stream Encapsulation (GSE)," Standard TS 102 606-1 V1.2.1, Jul. 2014.

[78] Consultative Committee for Space Data Systems, "Encapsulation Packet Protocol," CCSDS Blue Book 133.1-B-3, 2020.

[79] Smith, A., Brown, E. W., and Merat, F., "A Digital Testbed for Autonomous Spacecraft Communication Services," *IEEE Space Hardware and Radio Conference*, Las Vegas, NV, 2023.

[80] NASA GSFC, "Joint Polar Satellite System (JPSS) Level 1 Requirements," JPSS-REQ-1001/470-00031, Rev 2.1, Feb. 2019.

[81] Ashbury, S. C., "JPSS-1: Building the Nation's Next Generation Operational Polar-Orbiting Weather Satellite," *34th Space Symposium*, Colorado Springs, CO, 2018.

[82] NOAA Office of Satellite and Product Operations, "Satellite Alert Messages," ospo.noaa.gov/operations/messages, retrieved 21 August 2024.

[83] Roberts, B., Bester, M., Dumlao, R., Eckert, M., Johnson, S., Lewis, M., McDonald, J., Pease, D., Picard, G., and Thorsness, J., "Highly Automated On-Orbit Operations of the NuSTAR Telescope," *Observatory Operations: Strategies, Processes, and Systems V*, Vol. 9149, edited by A. B. Peck, C. R. Benn, and R. L. Seaman, International Society for Optics and Photonics, SPIE, 2014, p. 91490T. https://doi.org/https://doi.org/10.1117/12.2056547.

[84] NASA, "NISAR Mission Science Users' Handbook," JPL 400-1707, V1, Aug. 2019.

[85] Trowbridge, M., and Doubleday, J. R., "Intermediate Fidelity Solid State Recorder Modeling for NISAR," *10th International Workshop on Planning and Scheduling for Space*, Pittsburgh, CA, 2017.

[86] Sharma, P., Doubleday, J. R., Hammer, B. C., and Mohr, D., "Operations Concept for responding to Urgent Requests for NASA-ISRO Synthetic Aperture Radar (NISAR)," *IEEE Aerospace Conference*, Big Sky, MT, 2020.

[87] Mitola, J., *Cognitive Radio Architecture: The Engineering Foundations of Radio XML*, Wiley, Hoboken, NJ, 2006.