SSC24-IV-05

TechEdSat-11: Prototyping Autonomous Communications in Orbit

Rachel Dudukovich, Adam Gannon, Janette Briones, Timothy Recker,
Patrick Zhong,Kyle Deleon, Lucas Shalkhauser, Ethan Schweinsberg
NASA Glenn Research Center
21000 Brookpark Rd, Cleveland, OH
rachel.m.dudukovich@nasa.gov

Evan Danish
ZIN Technologies
6749 Engle Rd, Middleburg Heights, OH

Alejandro Salas
Millennium Engineering and Integration
1400 Crystal Drive, Suite 800 Arlington, VA

Marcus Murbach
NASA Ames Research Center
Moffett Blvd, Mountain View, CA

ABSTRACT

Cognitive Engine 1 (CE-1) is a state-of-the-art automated system designed to manage routine operations and respond to adverse events without requiring human operator input. CE-1 will optimize scheduling capabilities, detect and react to link failures, and ensure data delivery deadlines are met efficiently. Seamless roaming between government and commercial providers will allow for a robust and cost-effective network. TechEdSat-11 will demonstrate two key components of CE-1: User Initiated Services (UIS) and Delay Tolerant Networking (DTN). The first phase of the experiment will use UIS to automate on-demand scheduling of a commercial S-band ground station. Later phases of the experiment will demonstrate DTN store-and-forward and space internetworking capabilities through the TechEdSat-11 S-band radio. This paper will discuss the CE-1 main components relevant to the experiment, the flight and ground software architecture, experiment concept of operations and preliminary results.

1 INTRODUCTION

Cognitive engines for space communication have been a topic of interest to NASA since 2016. The ability to roam between service providers, autonomously reconfigure system parameters, optimize resource usage, and recover from unplanned events is increasingly relevant to the future space networks consisting of commercial service providers, as well various government organizations and agencies. Software defined radio (SDR) is a key enabling technology for several approaches to cognitive communications [1], as well as space internetworking [2], and on-demand scheduling [3]. The Cognitive Communications project at NASA Glenn Research Center has evaluated each of these main components and completed an in-depth analysis of the functional decomposition of a cognitive spacecraft communication system [4]. The result of this work has been a phased approach towards the design and implementation of Cognitive Engine 1 (CE-1). This paper discusses the implementation of the first phase of development, which is focused on establishing a fully automated spacecraft communication system, and the deployment of selected aspects of the CE-1 system on Technology Educational Satellite 11 (TES-11).

The remainder of this section provides an overview of the Technology Educational Satellite program and its work related to cognitive communications. Section 2 discusses the CE-1 system main components, including User Initiated Services (UIS) and Delay Tolerant Networking (DTN). Section 3 discusses the TES-11 CE-1 experiments concept of operations, success criteria, experiment flight software overview, and the ground system software pipeline and network. At the time of writing, TES-11 has passed all functional tests, was integrated onto a Firefly Alpha launch vehicle, and is scheduled to launch in June 2024.

1.1 Technology Educational Satellites

The TechEdSat series is a collaborative effort led by NASA Ames Research Center in partnership with various space agencies and educational institutions. The series primarily focuses on using the CubeSat platform for rapidly advancing nano-satellite technologies and capabilities, particularly those that improve satellite operations. Through its missions, the TechEdSat series has demonstrated the potential of small satellites to contribute to space research and to push the state of the art in a variety of experiments. Previous experiments on TechEdSat-13 were focused on demodulating the TES-13 S-band radio signal in Amazon Web Services (AWS) Ground Station, [5], flight testing an experimental neuromorphic processor, [6], and developing a specialized avionics stack for processing artificial intelligence applications on nano-satellites [7]. Building on the success of TechEdSat-13, the TES-11 mission will demonstrate advanced technologies for increased automation of end-to-end space communications.

2 CE-1 AUTOMATED SYSTEM

The CE-1 system consists of three main subsystems, with components on both the spacecraft and ground. Figure 1 shows a block diagram of the main CE-1 components. The data handling components are shown in the orange box, the service request system is shown in the green box, and the link management system is shown in the blue outlined box. The top portion of the diagram shows components on the user spacecraft and the bottom half shows the ground components.

CE-1 consists of NASA flight software components including High-rate Delay Tolerant Networking (HDTN) [8], User Initiated Services [9], Space Telecommunications Radio System (STRS) [10], and Network Toolkit [11]. Each of these components has heritage from on-orbit testing onboard the International Space Station and has been extensively tested on NASA GRC's Cognitive Ground Testbed [12]. As such, we believe the CE-1 system has significantly advanced the maturity of space communication system automation, beyond simulation and proof of concept studies. TES-11 will further this technology maturation by demonstrating the integration of aspects of the CE-1 system operating in orbit.

2.1 HDTN Introduction

HDTN manages the storage and transmission of user data (e.g., telemetry, science data, vehicle health). It is a high performance implementation of the main delay tolerant networking protocols that provide store-and-forward capabilities, reliable transport, routing, and security for space networks. HDTN was developed by NASA Glenn Research Center and is publicly available via the NASA Open Source Agreement [13], [14]. The main modules of HDTN are shown in Figure 2. HDTN was designed with a modular architecture, that scales to a wide range of missions including cognitive networks and CubeSat platforms [15]. The main interfaces to CE-1 are enabled using a series of application programming interface (API) commands. CE-1 uses the API to query HDTN's storage for data to transmit and responds with a contact plan that contains the schedule for service that was requested by UIS.

DTN contact plans are a concept developed in Contact Graph Routing [16]. Contact plans are a predetermined schedule of space communication links that describe the connectivity of the network at a given point in time using source, destination pairs, contact start and stop times, as well as data rates and distances between nodes. The contact plan is a central part of the CE-1 system. As additional contacts are scheduled, UIS generates new contact plans. HDTN consumes the new plan as an API message which is used to determine where to send data and whether data should be stored or transmitted at a given time.

A forward error correction application was developed for TES-11 to ensure files transmitted via DTN bundles can be reassembled even when a subset are lost [17]. TES-11's S-band radio supports downlink only. There is also a bi-directional low-rate command channel but it was not desirable to use this channel for data and acknowledgements. DTN typically uses Licklider Transmission Protocol (LTP) for reliable transport over space links, however it requires acknowledgements over a control channel or bi-directional link. DTN also provides a custody transfer mechanism but this approach has similar

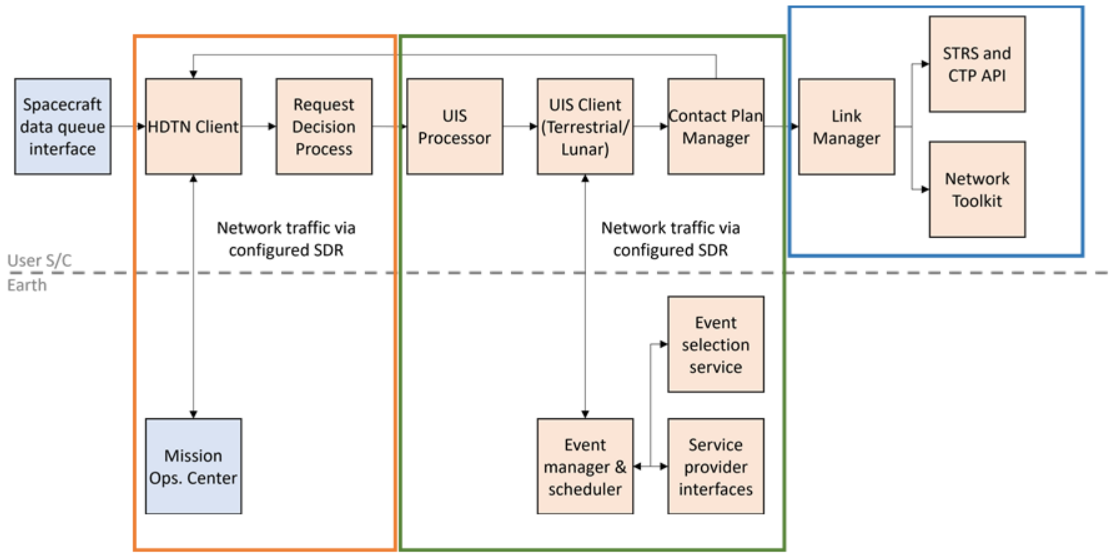


Figure 1: Cognitive Engine 1 Block Diagram

drawbacks to that of LTP. Both methods send a portion of data, check for it to be received, the receiver responds, and data is re-transmitted if it was not received. The forward error correction application is experimental but may provide an improvement over LTP and custody transfer.

2.2 UIS Introduction

UIS is a software framework that automates the process of a spacecraft scheduling communication service. It is capable of scheduling NASA's Tracking and Data Relay Satellite System (TDRSS), as well as commercial services such as AWS [9]. Figure 3 shows a high-level diagram of the UIS request and service fulfillment process. The spacecraft issues a generic request for service by specifying a data transmission volume and deadline. The request is sent over a low-rate control channel (TES-11 uses the Iridium Short Burst Data service) to the UIS server on Earth for scheduling. The UIS ground server processes the request and schedules service with providers using their specific APIs. The server responds to the spacecraft with an acknowledgement. The user spacecraft will begin to transmit data over the high-rate communication channel once the scheduled link is available.

The capabilities of UIS were extended for CE-1 to control multiple layers of the communication system. The UIS server generates a DTN contact plan which contains the source and destination endpoint ID associated with the scheduled link, link start and stop times in Unix Timestamps, and the radio waveform ID associated with the scheduled provider. IP routing associated with the scheduled provider are also configured. In this way , the contact plan is read by multiple elements of the communication system to orchestrate the dynamic configuration of the DTN layer, the network layer, data link layer and physical layer before data transmission begins.

3 EXPERIMENT

This section discusses the experiment success criteria and concepts, the overall CE-1 experiment design and development, specific details regarding the experiment flight software and ground system pipelines, and ground network. Figure 4 shows a block diagram of the TES-11 experiment concept of operations. TES-11 will host the HDTN and UIS experiment components separately and will be commanded to start several different experiment scripts throughout its operations. UIS will use both the Iridium control channel and S-Band data channel. HDTN will be commanded to start using the control channel but will primarily use the S-Band link to send generated DTN traffic (bundles) to the ground. Figure 5 shows the completed TES-11 satellite.

3.1 Experiment Success Criteria

The TES-11 experiment has been divided into four phases according to project priority and complexity. Initially, the UIS and HDTN experiments will be conducted separately, but will be combined for the final phases. The first phase of the experi-

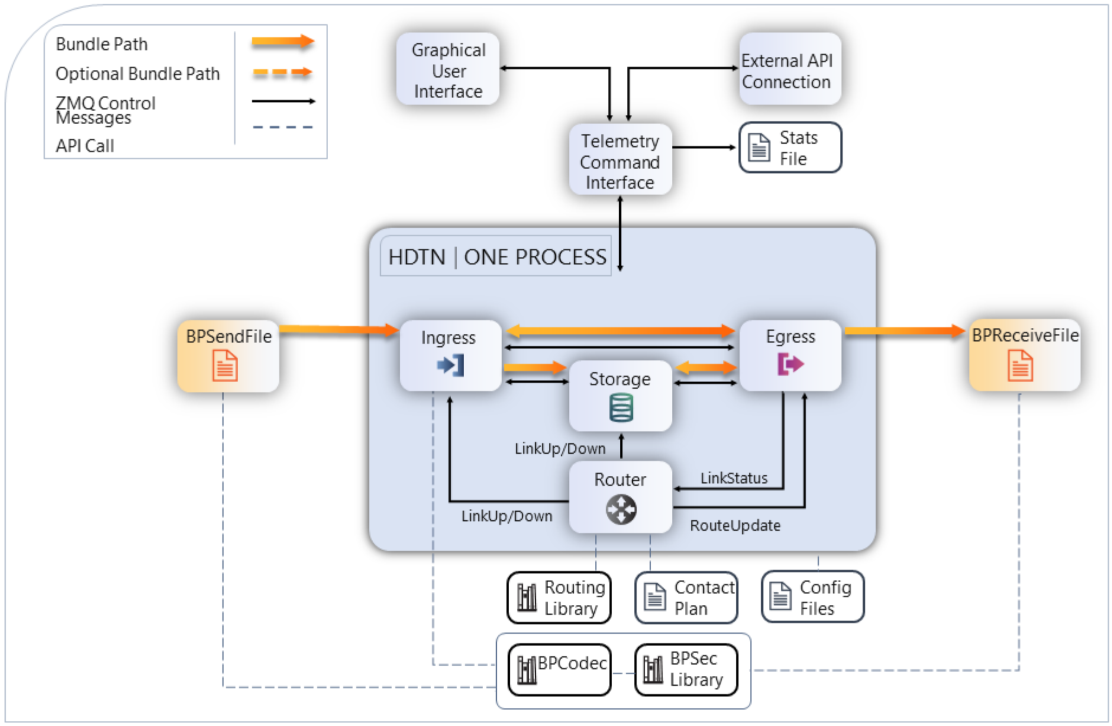


Figure 2: HDTN High-level Diagram

ment will focus on establishing UIS request capabilities. A UIS request message will be sent from TES-11 and received by the UIS Server at GRC which will simulate scheduling a contact. The test will be considered successful when logs confirm a response with correct start time makes it back to the spacecraft. In the second phase of the experiment, when UIS messages are sent from TES-11, the UIS server on the ground will schedule antenna time with AWS Ground Station. TES-11 will then start the S-band transmission of data at the proper time, and the signal will be received by AWS Ground Station. In addition, the second phase of the experiment will begin to incorporate HDTN. The spacecraft flight software will start up HDTN and command it to begin to transmit bundles via the spacecraft's S-band radio. Completion of Phase 2 will satisfy nominal project success criteria of the experiment.

Phase 3 experiments will build upon Phase 2 and establish full success once completed. TES-11 will simulate data generation which will cause multiple UIS messages to be sent, received, and scheduled with AWS. In the second portion of Phase 3, HDTN will send a series of bundles which should be correctly received at the AWS cloud. At this point, the UIS and HDTN experiments will still be operated separately.

Phase 4 experiments are considered stretch goals that would help to demonstrate the functionality of CE-1 but were not part of the initial TES-11 concept of operations. Phase 4 experiments will be conducted if Phase 3 is fully completed and there is time remaining to conduct additional experiments on TES-11. In this case, UIS will schedule a link as in Phase 3 but HDTN will flow bundles over the link instead of simulated data. This would demonstrate a portion of the overall CE-1 capabilities, although it is not the full CE-1 system. An additional goal of Phase 4 will be to incorporate the DTN Engineering Network (DEN)[18] on the ground. GRC hosts the DEN and has enabled an external connection between the DEN and Goddard's AWS Mission Cloud Platform. Once bundles are processed and received by AWS Ground Station, they will be passed to a DTN ground node hosted at the DEN. This will more realistically represent a final end-user receiving their mission science data from the DTN service.

3.2 Experiment Flight Software

The TES-11 platform consists of multiple interconnected boards. Two boards are used to carry out the HDTN and UIS experiments: the Lunar Radio board, which runs UIS software and handles downlink to the ground, and the Beacon And Mem-

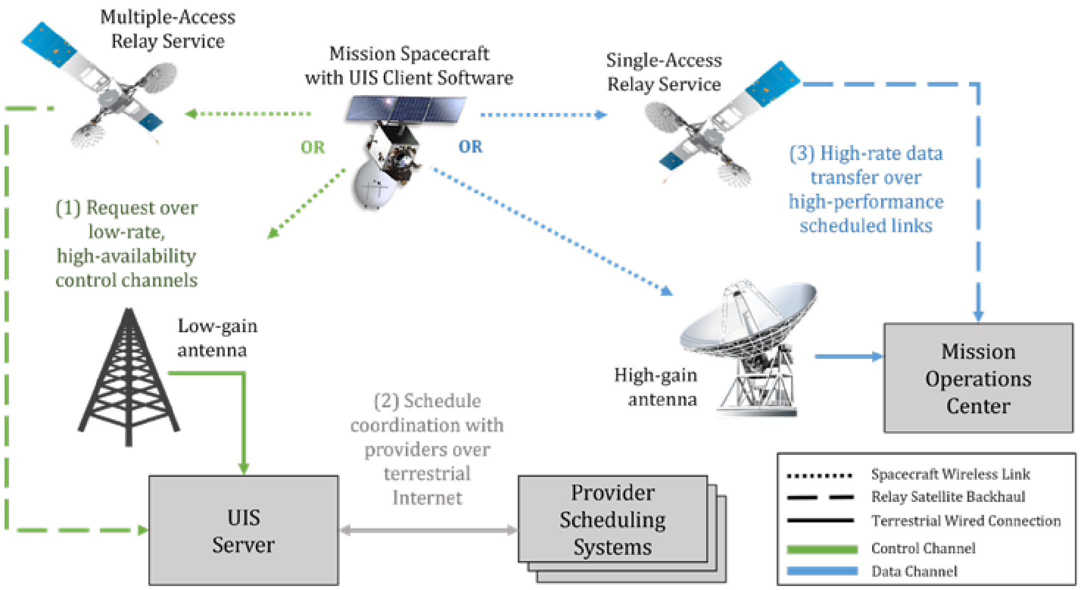


Figure 3: UIS High-level Diagram

ory Board Interface (BAMBI) board, which hosts HDTN and handles command and control. The BAMBI board interfaces with an Iridium modem. The Iridium modem supports bi-directional low-rate communication with the ground via the Iridium Satellite Network. It employs the short burst data (SBD) service to send short messages (a couple hundred bytes) between the ground and TES-11. SBD messages are used for command and control and for UIS requests and responses. The BAMBI board consists of a Portentia X8 processor running Linux and other components used for communication and experiments. The BAMBI board communicates with the Lunar Radio board over WiFi and UART. The Lunar Radio is an S-band SDR platform developed by the TES team from COTS parts. It is called the Lunar Radio after its potential capability to support a lunar CubeSat mission. The Lunar Radio board consists of an Ettus B205mini SDR and a Digi ConnectCore 6UL system-on-module (SOM) running Linux. The SOM is responsible controlling the SDR and providing it data to transmit. The Linux processors on both boards run control and experiment software. Figure 6 shows a block diagram of the main flight software components.

TES-OS, a custom-built software application for TechEdSat developed by NASA Ames, is used for command and control and experiment orchestration. It runs on both boards with the BAMBI instance of TES-OS controlling the Lunar Radio board (power on/off and commanding the Lunar Radio TES-OS). TES-OS manages communication between the two boards and seamlessly alternates between WiFi and UART communication as necessary. The WiFi communication channel supports higher throughput rates but does not function during S-Band transmissions. TES-OS starts and stops experiment software as commanded from the ground via SBD messages. It provides an interface for experiment applications to communicate between the two boards and the ground (over Iridium). Additionally, it supports "system administration" functionality such as reading/writing/deleting files on the filesystems and executing shell commands.

Data downlinked over S-band is QPSK modulated and encapsulated in CCSDS AOS frames [19]. AOS frames are Reed-Solomon encoded for error correction and scrambled. For the UIS experiment downlink, experiment data (read directly from on-disk files) is placed inside an AOS bitstream protocol data unit. For HDTN, bundles are wrapped in CCSDS encapsulation packets and sent as AOS multiplexing protocol data units. HDTN data is transmitted "live". That is, the downlink software receives bundles streamed from HDTN during the S-band transmission. During times when the transmitter is underfed, AOS Idle frames are sent to maintain the signal. For the UIS experiment downlink data, this is not necessary. Data to transmit is read directly from disk. The rate at which data is read is always higher than the transmit rate, so idle frames are never transmitted. The files are located on the

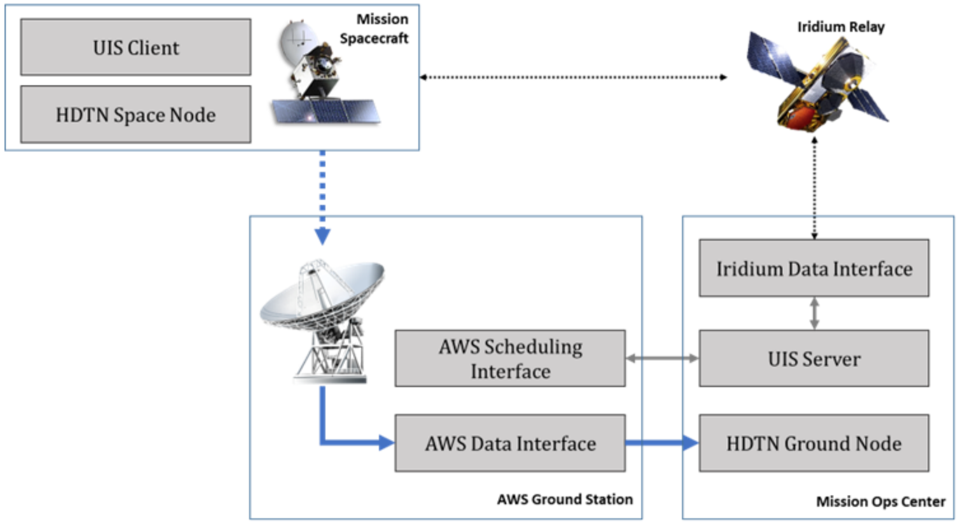


Figure 4: TES-11 High-level Concept of Operations

Lunar Radio filesystem. Their contents are controllable via ground commanding (e.g. system logs, sensor data, etc).

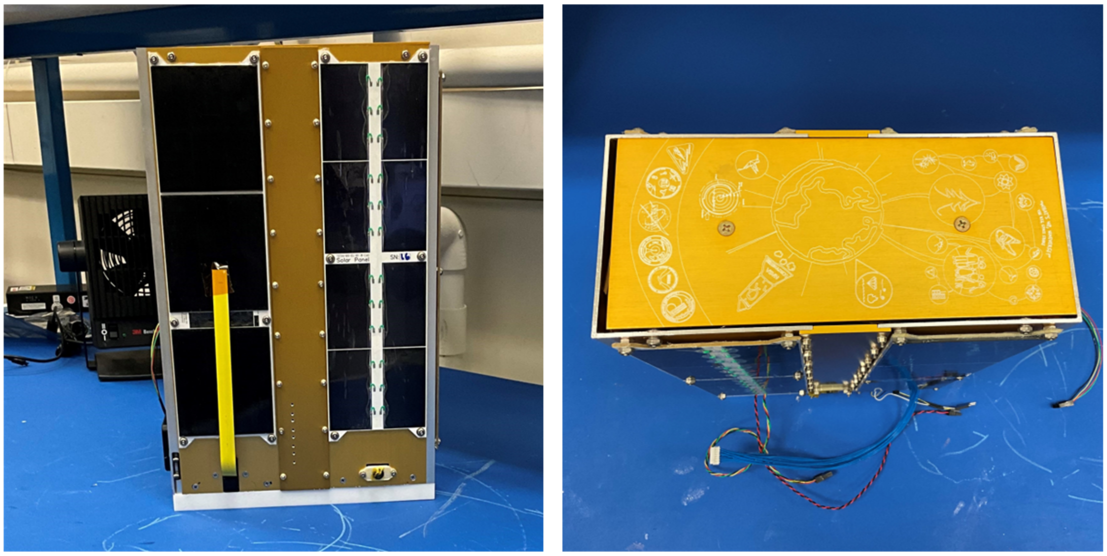
During the UIS experiment, the UIS flight software is automatically started on Lunar Radio boot. It communicates with the ground over Iridium through TES-OS. The UIS flight software issues a request to the UIS server running on the ground. A response message is sent back to TES-11 over Iridium. Upon receipt of the message, the flight software (assuming a contact has been scheduled) instructs TES-OS to start the transmitter and launch the downlink software at the specified contact time. At the end of the contact, TES-OS turns off the transmitter and stops the downlink software. During the contact, the downlink software reads the files to be transmitted from the filesystem, puts them in AOS frames as described above, and passes them to the Ettus SDR over a serial interface.

HDTN runs on the BAMBI board and is packaged inside a docker container. When started, the docker container automatically starts HDTN and and one of the HDTN apps used to generate bundles. The HDTN app is either bpgen or bpsendfile. Bpgen generates bundles at a fixed rate, while bpsendfile reads files off of disk. Both create bundles and forward them to HDTN. To carry out the HDTN experiment, TES-OS simultaneously starts the HDTN docker container on the BAMBI board and the downlink software on the Lunar Radio. The HDTN docker container generates bundles using either bpsendfile or bpgen and forwards them to HDTN, also running in the docker container. The HDTN instance immediately forwards received bundles to TES-OS, which transfers them to the Lunar Radio board and delivers them to the downlink software. The downlink software then encapsulates and encodes the bundles for transmission as described above. The signal is passed to the Ettus SDR and transmitted to the ground.

Two downlink "modes" have been described above: one for sending AOS frames containing the contents of on-disk files, the other for sending AOS frames encapsulating bundles from HDTN. The HDTN experiment always uses the latter mode. The mode for the UIS experiment is configurable. Initial testing will be carried out using the file-based downlink mode. During testing of the stretch-goal experiment described in the previous section, when UIS and HDTN operate together, the bundle downlink mode will be used. After UIS schedules the contact, at contact start TES-OS will launch the HDTN docker container and start the bundle downlink software. Bundles will be transmitted to the ground for the duration of the automatically scheduled contact.

3.3 Ground System Pipeline

Before data can be downlinked from TES-11 to the ground system, a contact must be scheduled. The scheduling request includes: the satellite, ground station, and a mission profile. A mission profile sets configurable parameters in AWS Ground Station which include antenna configuration, tracking configuration, settings for contact pre-pass, pass, and post-pass time windows, and definition of a des-



A) TES-11 CubeSat B) TES-11 Top View

Figure 5: Completed TES-11 CubeSat

tination endpoint. In this case, the destination endpoint is configured for the subnet ID and IP address of an EC2 instance since the goal is to stream and process downlink data live rather than store it, e.g. in an S3 bucket. Once all these mission parameters are specified and the satellite's catalog number is provided to AWS, contacts can be manually scheduled in the AWS console for the Phase 3 and 4 HDTN experiments. Figure 7 shows a block diagram of the Phase 4 data pipeline.

During an active contact between TES-11 and AWS Ground Station Ohio, the signal received at the ground station is digitized and encapsulated into a VITA Radio Transport (VRT) packet stream [20], which is delivered over an Ethernet interface to the ground network, where the stream is processed by Data Defender and forwarded via UDP to a particular EC2 host on the network. Data Defender is an application that provides a lossless and encrypted data pipe between the Ground Station antenna and an EC2 instance; streaming data to an EC2 endpoint without Data Defender is unsupported on AWS Ground Station, thus the use of Data Defender and VRT decapsulation is required. On the EC2 instance a software data processing pipeline receives the UDP packets from Data Defender and deframes each layer to retrieve the original data payload from TES-11. This is done through a series of Unix sockets and pipes in the following order: VRT decapsulation, demodulation of the IQ samples received at the antenna, Reed-Solomon decoding, AOS frame decoding, and finally CCSDS de-encapsulation.

This leaves only Bundle Protocol version 7 bundles, which are forwarded via UDP to an HDTN process running on the EC2 instance for final routing and forwarding. Since the data payload in the bundles consists of files, the final destination for all data transmitted by TES-11 is a bpreceivefile application which recovers all files sent by the satellite and saves them to a local storage device attached to the host, reconstructing as necessary any files that may have been fragmented during transmission. The bpreceivefile application can run on the cloud (Phase 3), on the DEN extranet, or on a host within any of the networks connected to the DEN extranet (Phase 4).

UIS handles automatically scheduling ground contacts (as opposed to the scheduling process used with HDTN, described previously). For UIS experiments using S3 data delivery, the packet captures delivered to the S3 buckets are verified after the pass is completed. The AOS frames are recovered from the signal and from the bitstream in those frames, the files are reconstructed. The UIS experiment can also be run using a data pipeline similar to that of HDTN. Here, EC2 delivery is used. The pipeline is the same as that for HDTN up until the point at which data is extracted from the AOS frames. The AOS frames for the UIS experiment contain bitstream data. This bitstream contains the contents of the files transmitted during the experiment. This bitstream is written directly to disk.

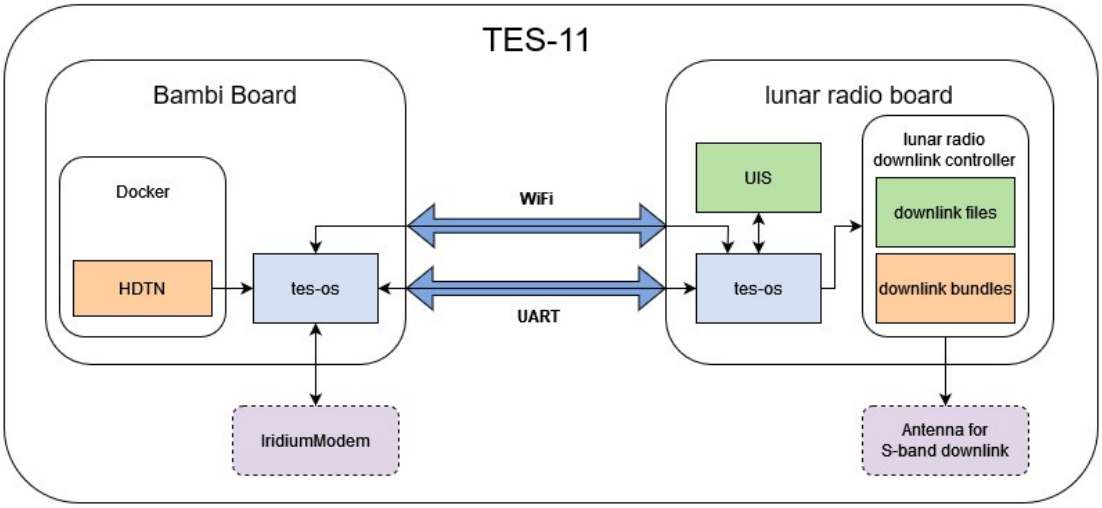


Figure 6: Flight Software Architecture

3.4 Ground Network

The ground network consists of two private subnetworks within the Cognitive VPC (C-VPC) hosted on MCP AWS, plus the DEN extranet, which is connected to the C-VPC through a site-to-site Virtual Private Network (VPN) tunnel. When downlink data reaches a host on the DEN in Phase 4, it can be processed locally or forwarded throughout the DEN extranet. This extranet includes additional active VPN connections to an HDTN VPC (H-VPC) similar but distinct from C-VPC and also to the external lab networks of partners such as Goddard Space Flight Center, Marshall Space Flight Center, Johnson Space Center, and Jet Propulsion Laboratory. Figure 8 shows a network diagram of the DEN connection to AWS Ground Station.

In the primary version of Phase 4 of the HDTN experiment, the two hop architecture from satellite, to ground station network, to the DEN lab network is designed to emulate a mission scenario wherein a satellite or spacecraft downlinks science data through a gateway to a mission operations center. Once the science data is delivered to the "end user" at the DEN, it can be processed or further transferred at their discretion using terrestrial networking protocols. However, the many hosts, services, and peer networks attached to the DEN extranet afford the flexibility to model alternative mission scenarios and architectures in the future.

4 FUNCTIONAL TESTS

Prior to flight vehicle integration the software components were tested as much as possible. An engineering model of the satellite was built and located on a roof at NASA GRC – providing visibility to Iridium satellites for end-to-end tests through the network. This engineering model consisted of a Lunar Radio board, a stand-in computer (an Intel NUC) acting as the BAMBI board, and an Iridium Modem. The SDR output on the Lunar Radio board was connected via coaxial cable to a receiver SDR in the lab. A full end-to-end test was not possible to complete since TES-11 is not able to send a real signal to AWS Ground Station.

Two ground tests were performed to verify HDTN functionality. First, using the engineering model, SBD messages were sent to command the model satellite to start the HDTN experiment. The satellite output a valid QPSK signal as verified with a spectrum analyzer. This signal was then passed through the demodulation/decode pipeline in the lab which produced valid bundles containing the experiment data from HDTN.

The second test was performed using the TES-11 satellite before delivery to the launch provider. The satellite was placed in line-of-sight to the sky, and the HDTN experiment was started via SBD command messages. It was verified that the S-band transmission by the satellite was a valid QPSK signal. Due to limitations with the test setup, the demodulation/decode pipeline could not be run to further verify the signal.

UIS functional testing was completed on April 22

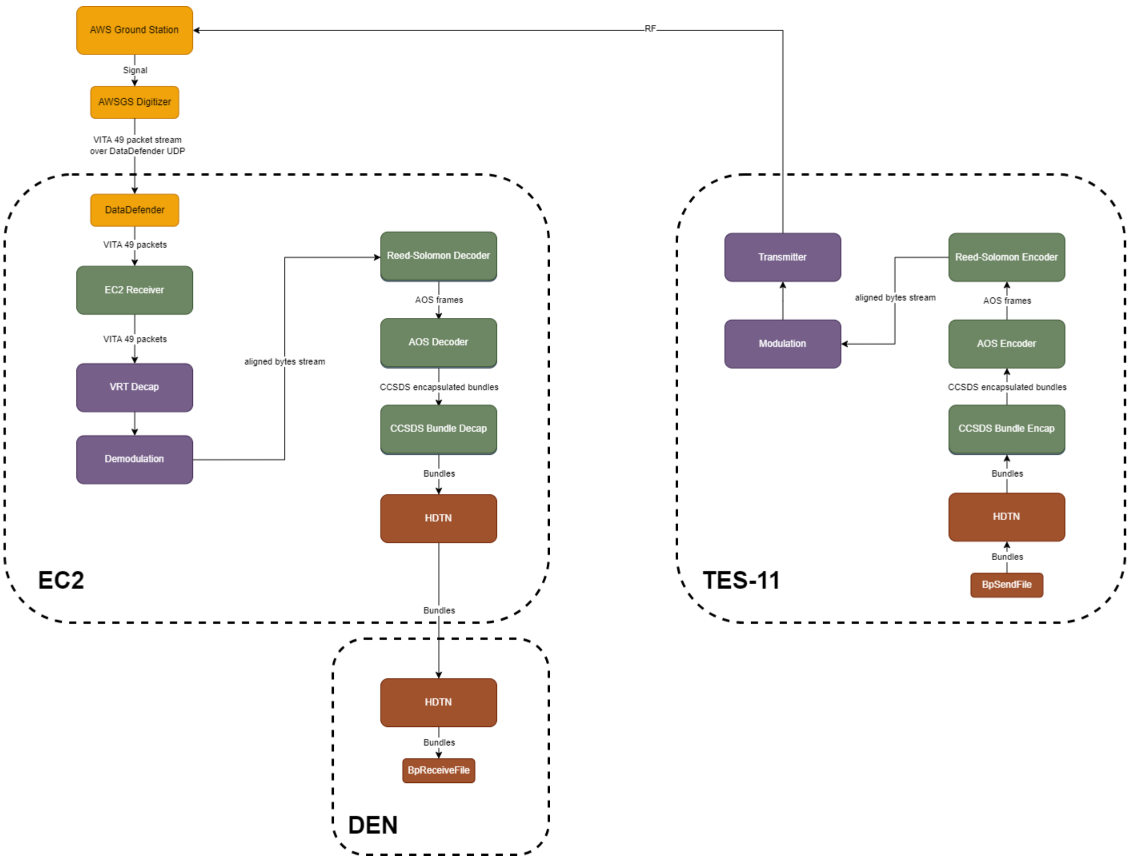


Figure 7: Complete Phase 4 Data Pipeline

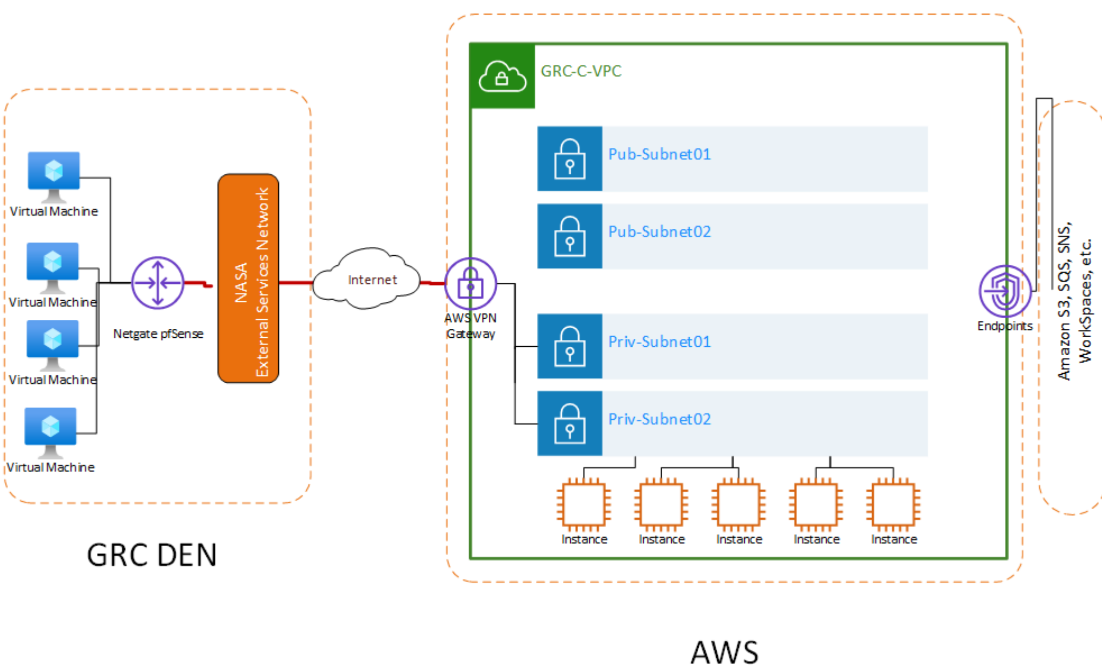


Figure 8: AWS Cognitive VPC with VPN connection to the DEN.

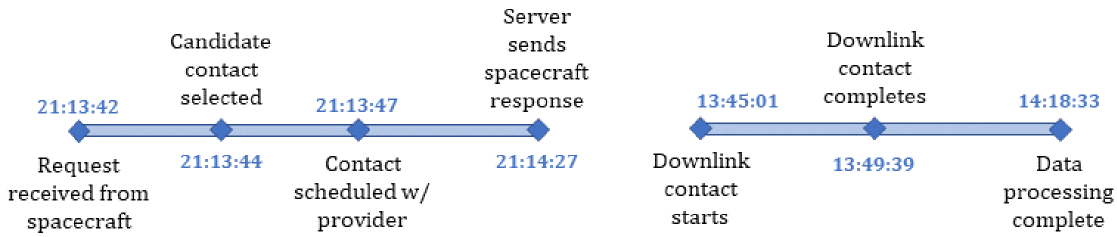


Figure 9: Timeline of UIS contact scheduling and service execution.

to 23, 2024 using the engineering model. The Iridium SBD service was used for the UIS commands and responses. The TES-11 avionics boards, UIS ground server, and AWS Ground Station were utilized as they will be in flight. Figure 9 shows a timeline of events. UIS scheduling was tested on the ground as follows:

- UIS on the TES-11 Lunar Radio board issues a request which was received by the UIS server at NASA GRC.
- The server analyzes possible contact times available between TES-13 and the AWS Ground Station in Ohio that will meet the data volume and deadline requirements specified in the UIS request. The two-line element (TLE) for TES-13 was used as a substitute for TES-11, since TES-11 was not in orbit at the time of testing.
- Once a contact is found, it is confirmed with AWS Ground Station and a positive response is sent to the TES-11 avionics. The contact was scheduled for early the next day (April 23) due to AWS availability.

As with HDTN, the main difference from flight during this test is that RF signals from the Lunar Radio cannot flow through AWS Ground Station.

The AWS Ground Station mission profile has also been tested using the satellite catalog number and contact schedule for the previously launched TES-13, verifying successful delivery and capture of VITA 49 packets to the target EC2 endpoint.

5 CONCLUSION

The partnership between the TechEdSat program at Ames Research Center and the Cognitive Communications project at NASA Glenn Research Center (GRC) continues to advance communications and avionics technology innovations for nano-satellites. TES-11 will build upon the success of TES-13 and will demonstrate several additional capabilities. The user initiated services experiment will demonstrate on-demand scheduling of a commercial service provider by a small user spacecraft. Following that, the HDTN experiment will demonstrate modern delay tolerant networking techniques including Bundle Protocol version 7 and reliable transport via forward error correction. The final phases of the experiment will demonstrate integration of UIS and DTN by flowing DTN traffic over the dynamically scheduled link. The experiment features a complete demodulation and decoding pipeline based on AWS Ground Station with integration into the DTN Engineering Network. For these reasons, we believe TES-11 significantly advances the state of machine-to-machine communications, communications automation and network management, and integration of government and commercial networks. Further, this work outlines the path forward to enhance communication system automation by providing a detailed decomposition of the network decision-making elements.

Acknowledgments

The authors would like to thank the NASA Space Communications and Navigation (SCaN) program, and in particular Jason Mitchell (NASA/HQ) for supporting this work.

References

[1] Timothy M. Hackett et al. “Implementation of a space communications cognitive engine”. In: *2017 Cognitive Communications for Aerospace Applications Workshop (CCAA)*. 2017, pp. 1–7. DOI: 10.1109/CCAAW.2017.8001607.

[2] E. Jay Wyatt et al. “Autonomous Networking for Robotic Deep Space Exploration”. In: *International Symposium on Artificial Intelligence, Robotics, and Automation for Space*

(ISAIRAS 2018). Madrid, Spain, July 2018. URL: `https://ai.jpl.nasa.gov/public/papers/wyatt-isairas2018-networking.pdf`.

[3] Timothy M Hackett et al. "Demand Access for NASA's Deep Space Network: A New Paradigm for Operations". In: *16th International Conference on Space Operations (SpaceOps)*. Virtual, May 2021.

[4] D. Chelmins et al. "Cognitive Communications for NASA Space Systems". In: *International Communications Satellite Systems Conference (ICSSC)*. 2019.

[5] Adam Gannon et al. "Cloud-Based Demodulation and Data Distribution of a Satellite Downlink". In: *2023 IEEE Cognitive Communications for Aerospace Applications Workshop (CCAAW)*. 2023, pp. 1–5. DOI: `10.1109/CCAAW57883.2023.10219312`.

[6] Nayim Rahman et al. "Neuromorphic Hardware in Outer Space: Software Defined Networking Executed on an In-Orbit Loihi Spiking Processor". In: *2023 IEEE Cognitive Communications for Aerospace Applications Workshop (CCAAW)*. 2023, pp. 1–5. DOI: `10.1109/CCAAW57883.2023.10219299`.

[7] M. Murbach et al. "BRAINSTACK – A Platform for Artificial Intelligence Machine Learning Collaborative Experiments on a Nano-Satellite". In: *37th Annual Small Satellite Conference*. 2023.

[8] R. Dudukovich et al. "Advances in High-rate Delay Tolerant Networking On-board the International Space Station". In: *IEEE Space Mission Challenges for Information Technology - IEEE Space Computing Conference* (July 2024).

[9] Adam Gannon et al. "Spacecraft-Initiated Scheduling of Commercial Communications Services". In: Oct. 2022. DOI: `10.1049/icp.2023.1394`.

[10] Richard Reinhart et al. "NASA's Space Communications and Navigation Test Bed aboard the International Space Station". In: *Aerospace and Electronic Systems Magazine, IEEE* 28 (Apr. 2013), pp. 4–15. DOI: `10.1109/MAES.2013.6506824`.

[11] David Brooks et al. "In-Space Networking On NASA's SCaN Testbed". In: Oct. 2016. DOI: `10.2514/6.2016-5754`.

[12] Joseph Downey et al. "Emulated Spacecraft Communication Testbed for Evaluating Cognitive Networking Technology". In: *CCAAW'23: IEEE Cognitive Comms. for Aerospace Applications Workshop*. 2023.

[13] S. Booth et al. *High-Rate Delay Tolerant Networking (HDTN) User Guide Version 1.3*. Technical Memorandum. NASA, 2024. URL: `https://ntrs.nasa.gov/citations/20230015434`.

[14] NASA Glenn Research Center. *High-Rate Delay Tolerant Network*. URL: `https://github.com/nasa/HDTN`.

[15] Rachel Dudukovich et al. "Microservice Architecture for Cognitive Networks". In: *2020 IEEE International Conference on Wireless for Space and Extreme Environments (WiSEE)*. 2020, pp. 39–44. DOI: `10.1109/WiSEE44079.2020.9262617`.

[16] S. Burleigh. *Contact Graph Routing*. 2009. URL: `https://datatracker.ietf.org/doc/html/draft-burleigh-dtnrg-cgr-00` (visited on 06/08/2024).

[17] Noah Douglass et al. "Application of Fountain Code to High-Rate Delay Tolerant Networks". In: *IEEE Access* PP (Jan. 2023), pp. 1–1. DOI: `10.1109/ACCESS.2023.3315659`.

[18] B. LaFuente et al. "Exploring New Frontiers in Space Communications: Enhancing Delay Tolerant Networking through Cloud and Containerization". In: *AIAA SCITECH 2024 Forum* (2024). URL: `https://api.semanticscholar.org/CorpusID:267474202`.

[19] *AOS Space Data Link Protocol*. Standard CCSDS 732.0-B-4.

[20] *Vita Radio Transport (VRT)*. Standard ANSI/VITA 49.0-2015.