AUTOMATED END-TO-END SPACECRAFT CONNECTIVITY ACROSS DIVERSE LINKS

Adam Gannon[1*], *David Brooks*[2], *Shadi Zogheib*[1], *Ethan Schweinsberg*[1], *Lucas Shalkhauser*[1], *Brian Frantz*[3], *Mick Koch*[1], *Rachel Dudukovich*[1], *Joseph Downey*[1], *and Janette Briones*[1]

[1]*NASA Glenn Research Center, Cleveland, Ohio, USA*
[2]*SAIC, Inc., 12010 Sunset Hills Rd, Reston, VA 20190*
[3]*HX5, LLC., 2001 Aerospace Parkway, Brook Park, OH USA*
[*]*adam.gannon@nasa.gov*

Keywords: SATELLITE COMMUNICATION, AUTOMATION, INTEROPERABILITY, ROAMING

Abstract

An increasing variety of communications services are available to space missions. Yet varying standards between providers hinder adoption due to the complexity of managing configurations for each communication pathway. We present a software framework which alleviates this issue by automatically establishing end-to-end communications during contacts with a provider. The proposed development configures spacecraft protocols at the physical, link, and network layers. The spacecraft remains reachable at the same IP address regardless of the active provider. Underlying protocols and routes are abstracted, allowing the user to simply send data to a destination with the framework ensuring its delivery. We evaluate a full implementation of the framework in laboratory experiments conducted on an emulated communications testbed. These tests demonstrate data delivery across three different services with rapid (<20s) reconfiguration as the spacecraft transitions between providers.

1 Introduction

Science and exploration missions of the future will benefit from a heterogeneous mix of space communications services offered by a variety of providers [1]. There is growing interest in terminals which can seamlessly interoperate between many providers, allowing the spacecraft to use the best service for its data transfer needs [2, 3]. This vision is challenged by unique and often times incompatible protocol sets used by each provider. Though adoption of fifth generation (5G) non-terrestrial network provisions [4] has the potential to harmonize future networks, spacecraft aiming for interoperability with existing providers must contend with a variety of standards. These include specifications for network addressing, packet formatting, modulation, frequencies, and more. Manual management of these parameters for every class of link places a burden on mission operations which grows with the number of providers it attempts to use. Without a high degree of automation it will be difficult for a spacecraft to take full advantage of diverse services to best meet its specific needs.

We note that the vast majority of providers support Internet Protocol (IP) traffic across their services. This is particularly true for commercial relay providers whose primary market is delivery of Internet connectivity to terrestrial customers. With our proposed software, higher-layer protocols will flow application data to end destinations regardless of which provider service is active at any given time. Indeed, provided a network flow can meet quality of service requirements for data delivery, the spacecraft operator likely has little preference for which services or intermediate network nodes the traffic traverses. Our approach will maintain a persistent spacecraft IP address. This is conceptually similar to terrestrial Mobile IP [5] which is infrequently supported in satellite provider networks where relay mobility causes frequent handovers [6]. Leveraging the deterministic nature of scheduled contacts, our system eliminates the handshaking a Mobile IP approach would require.

In this work, we propose an Automatic Service Execution (ASE) framework which establishes IP flows across various service provider networks. Software on the spacecraft and a ground-based server handle configuration of lower-layer protocols without explicit user input. The framework will switch between contacts as a spacecraft orbits, starting and stopping each according to times scheduled with the provider. Especially when combined with automated service scheduling [7], this development produces a similar experience for spacecraft to that of a terrestrial mobile broadband user whose cell phone can roam between base stations operated by different providers. Proposed software is evaluated in a testbed which emulates a spacecraft in low-Earth orbit transferring data to Earth over contacts switched between three service providers. Across dozens of contacts, we demonstrate automatic link configuration and transfer of data to its destination.

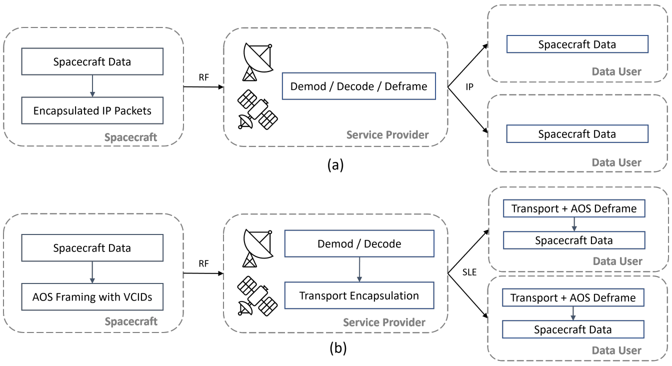

Figure 1: Comparison between routing of (a) Internet-routable IP packets and (b) AOS-framed raw data.

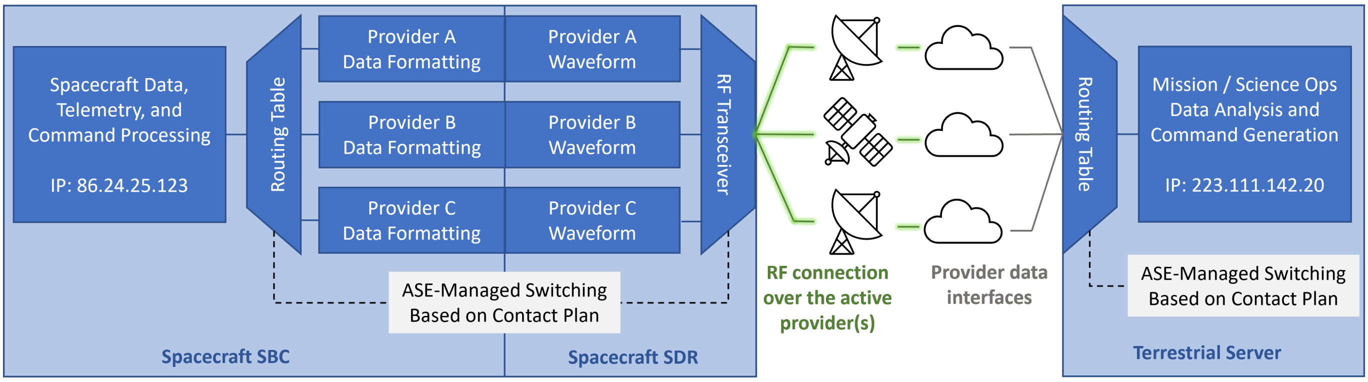

Figure 2: High-level diagram of the ASE framework with pseduo-IP addresses. The software sets the proper route and protocol set based upon the active provider. Data between spacecraft and operations center traverses RF and the terrestrial Internet.

2 A Common Layer for Interoperability

Spacecraft with information (science data, vehicle health, etc.) destined for Earth transmit signals which are received by ground stations – either directly or via relay satellites. Information decoded at these sites is typically sent over terrestrial Internet to mission operators, users of science data, and other interested parties. Spacecraft commands also flow from the mission operations center to the spacecraft. Since space links are highly asymmetric in favor of communications to Earth, this work will focus on the downlink without loss of generality. To operate with a provider's system, a spacecraft must transmit with physical-layer parameters (frequency, modulation, symbol rate, etc.) and packet formatting compliant with the provider's standards.

Modern relay satellites primarily serving users on Earth largely focus on broadband data delivery as a replacement for fiber or cable. To meet this customer need, these services have adopted IP-compatible protocols throughout their networks including data delivery to end users [8, 9]. Spacecraft use of these IP-native services requires compliant protocols for data routing to a mission operations center (Fig. 1a). Recent ground station as-a-service offerings tout internet connectivity as a selling point – frequently locating these in close proximity to data centers. If proper protocols are used, data received by these ground stations can be directly routed to destination(s) over the terrestrial Internet.

In contrast, many spacecraft and the infrastructure built to support them use the Advanced Orbiting Systems (AOS) specification [10] developed by the CCSDS standards body largely before worldwide adoption of IP [11]. Raw data from spacecraft subsystems can be embedded into AOS *transfer frames* with virtual channel IDs corresponding to different data types. In this case, frames received from a spacecraft are not directly routable over the terrestrial Internet. Mechanisms such as CCSDS Space Link Extension (SLE) can be used to forward non-routable frames from remote ground stations over the terrestrial Internet but this requires deframing software at each data destination (Fig. 1b). Alternatively, CCSDS defines a standard for IP over space links [12] the implementation of which will be described in Section 5.

To interoperate with a mix of systems, we use IP as the lowest common layer in the ASE framework. In addition to compatibility with IP-native providers, this allows use of standard networking tools on the spacecraft and ground systems for routing of data. Common terrestrial applications (e.g. file transfer, voice-over-IP) can be used directly in this IP-based system. This approach also facilitates the use of Delay Tolerant Networking (DTN) – a protocol suite for data transfer among space assets with intermittent connectivity [13]. DTN application software uses convergence layers to transfer groups of user data ("bundles") over IP. Finally, IP allows modern security (IPsec) protocols to be applied to space links [14, 15]. This can fulfill encryption requirements, such as NASA's mandate that all commands to spacecraft be encrypted [16].

3 Software Components

Figure 2 illustrates the main processes of the ASE framework. Data is generated on the spacecraft destined for an IP-addressable mission operations center (MOC) with a IP-addressable destination (`223.111.142.20` in the diagram). This traffic will be sent through a provider's systems during an active contact. Each provider may require different physical, link, or network layer formatting to operate with their system. The ASE framework will configure provider-compatible protocols during each contact. Data-generating spacecraft subsystems need not track intermediate hops – only send data to the destination IP. Likewise, in the uplink, ground systems must know through which provider the spacecraft can be reached. Uplink data is sent to a common destination IP (`86.24.25.123` in the diagram) regardless of which provider is active, and the ASE framework will handle the intermediate routing.

Two major software components are required to accomplish this connectivity: (i) physical-layer waveforms which run on the radio and (ii) network interface devices on the spacecraft processor which provide data framing and link addressing. Section 4 will describe the control of these components during a contact.

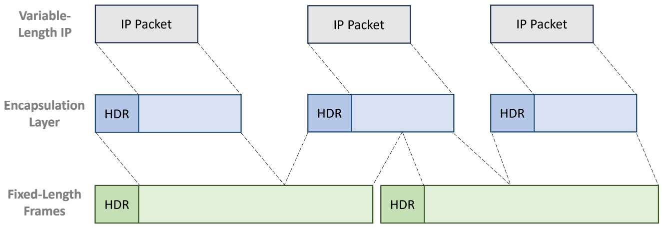

Figure 3: General method of encapsulating variable-length IP into fixed-length frames used in a communications standard A header ("HDR") is added to each layer.

Parameter	Description	Example
Source	Node address of data sender, in IPN scheme.	10
Destination	Node address of data receiver, in IPN scheme.	11
Start	Beginning of contact availability, as Unix timestamp.	1727343000
End	End of contact availability, as Unix timestamp.	1727344200
Data Rate	Anticipated link capacity used for CGR calculations, in bits per second.	50,000
Radio ID	Identifier for radio which is compatible for the given configuration.	“TTC-SDR”
Priority	Used to configure primary/secondary routes to the same destination.	1
Configuration ID	Unique identifier of provider and its configuration parameters (e.g. frequency, rate).	“TDRS50KMA”

Table 1: Contact plan parameters with example values. Standard parameters used in contact graph routing are above the line. Additional parameters added for ASE management of wireless links are described below.

3.1 IP-Compatible Radio Links

Each protocol used by a provider will have a standardized mechanism of embedding IP datagrams into the modulated RF signals sent over a wireless link. Standards-compliant modems deployed at a provider site will be able to decapsulate these packets and extract routable IP traffic. Frequently, the lowest-layer framing in these standards is of a fixed length to allow for frame synchronization and provide a suitable block size for forward error correction. Mapping of variable-length IP datagrams into these frames is typically accomplished by one or more encapsulation layers as in Fig. 3.

As in [2], we assume the spacecraft communications terminal is based on a software-defined radio (SDR) to allow rapid reconfiguration to support provider protocols with the same hardware. In Section 5 we discuss specific SDR protocol implementations (“waveforms”) created for testing. However, this concept equally extends to multiple hardware-defined terminals, each exclusive to an individual service.

3.2 Network Interfaces For Packetization

In addition to physical protocols, service providers implement standards to address requirements such as security, link scheduling, billing, quality of service, and terminal authorization/authentication services. These define what a spacecraft requires to exchange IP datagrams with the provider’s network. In contrast to early mission software which was written on low-level microcontrollers, many modern missions requiring complex computing capabilities fly single-board computers (SBCs) with full operating systems [17]. The ASE framework leverages the Linux network protocol suite to send IP datagrams to an end user by specifying peer’s destination address.

Linux kernel network interface devices implement physical data framing and link addressing standards to exchange packets with other local subnet peers. Additionally, these interfaces are assigned one or more IP host and network gateway addresses to define reachable local subnet IP peers using physical subnet link addressing. Linux routing uses the local subnet hardware addresses, IP gateway addresses, and routing table route priority metric to determine the output network device that can either deliver the datagram directly to the destination peer or “next hop” router with access to additional subnets.

The ASE framework provides an IP gateway for data from spacecraft subsystems to reach an IP-addressable ground destination during a scheduled contact with a flight network interface IP gateway address assigned on the provider’s ground station subnet. The provider routes space-originating IP datagrams, through the Internet, to the destination IP ground subnet. The converse is true when a ground application sends an IP datagram to the spacecraft SBC’s address.

4 System Orchestration

4.1 Contact Plan

Components allowing end-to-end connectivity must be configured as the spacecraft switches the active provider it communicates with. For this control, the spacecraft requires an exhaustive list of all contacts it has available to use. This list must be updated in real-time as additional service is coordinated with providers and the updated contact plan distributed to the spacecraft. Each scheduled contact implies the provider will be able to route spacecraft data through its internal network at that time.

The ASE framework leverages the contact plan required for contact graph routing (CGR) in DTN [18]. Each contact in the plan is defined by a start/stop time and includes an InterPlanetary Network (IPN) node number [19] which HDTN software maps to an IP address. The contact plan is used by CGR to determine the next-hop route. To this the ASE framework adds several fields for link configuration as detailed in Table 1.

4.2 Link Switching

The top-level process of the ASE framework is described in Algorithm 1. For each contact in the plan which is beginning, the ASE framework onboard the spacecraft will take the following actions:

1. Configure the selected radio with compatible physical-layer protocols for the active provider and command it to start transmitting.
2. Set the corresponding network interface up and adjust the default route so traffic towards the destination(s) flow over that interface.
3. Append contact information including end time to an internal list of currently-active contacts.

This process takes place a configurable t_{lead} seconds before the contact start to allow time for radio configuration. Occasionally, contacts may be extended by negotiating more time from the provider. The framework will extend the time it keeps a contact active if the end time in the contact plan changes. Otherwise, it will stop each link after it has been active for its planned duration. This stop process ceases radio transmission and sets the corresponding network interface down.

Each spacecraft has a gateway machine on Earth (either in the cloud or at a network operations center) which routes communications terrestrially on behalf of the spacecraft. Traffic is sent to the spacecraft via this gateway which also has an up-to-date contact plan for knowledge of which active links the spacecraft is can be reached through. The ASE framework software on this gateway machine also operates from a contact plan and switches routes to the terrestrial data interface of the active service provider. This process follows similar steps as in Algorithm 1 except the radio reconfiguration is not necessary.

Algorithm 1 Pseudocode of system orchestration loop. Steps which occur only on the spacecraft are indicated with †.

```
$actives \leftarrow \{\}$
while True do
    Initialize $plan$ from contact plan (Table 1).
    ▷Stop contacts if we're past their end times.
    for $active \in actives$ do
        if $active.end <$ now then
            Command transmit stop to selected radio†.
            Set network interface link down.
            $actives$.pop($active$)
        end if
    end for
    ▷Adjust stop of existing contact, or start new ones.
    for $contact \in plan$ do
        if $contact \in actives$ then
            $actives.end \leftarrow contact.end$
        else if $contact.start >$ (now + $t_{\text{lead}}$) then
            Command configuration to radio and start†.
            Configure network interface and set link up.
            $actives$.append($contact$)
        end if
    end for
end while
```

5 Testbed Implementation

The framework is evaluated on an emulated spacecraft communications testbed [20] at NASA Glenn Research Center in Cleveland, Ohio (Fig. 4). Space-side software is deployed on SDRs and SBCs which emulate small satellite hardware. Service provider hardware is emulated with three rackmount modems: Teledyne Paradise Datacom Qflex-400, Kratos OpenSpace quantumRadio, and Amergint satTRAC. Where necessary, custom software translates packet outputs from these modems into routable IP, standing in for similar systems which would be deployed inside a provider's ground network. RF connections between the spacecraft and modems pass through a Keysight Propsim channel emulator which makes/breaks connections based on the scheduled service. Ansys Systems Tool Kit (STK) is used to model the environment of the scenario under test. From orbital mechanics and RF transceiver properties, we derive key metrics such as carrier-to-noise density ratio (C/N_0), Doppler shift, and time delay of received signals. Using this input, the Propsim adds these impairments to signals such that they reach testbed receivers with similar signal conditions as would be experienced on orbit.

A Linux kernel virtual machine is used for the gateway (GW) through which traffic to/from the spacecraft flows. Software on this machine implements the ASE orchestration functionality to connect the data interface of the active emulated provider(s). Destination for spacecraft data is a MOC emulated with another virtual machine. The MOC can also generate commands for the spacecraft, though our testing will focus on the downlink.

Two CesiumAstro SDR-1001s are used to emulate a typical spacecraft radio configuration. SDR A is used primarily for scheduled high-rate data contacts to provider systems. The ASE framework is extensible to an arbitrary number of standards, even proprietary protocols, but open standards are used in the testbed for ease of implementation. SDR A can be reconfigured with waveforms* supporting CCSDS or the Second Generation Digital Video Broadcasting – Satellite (DVB-S2) standard which will be discussed below.

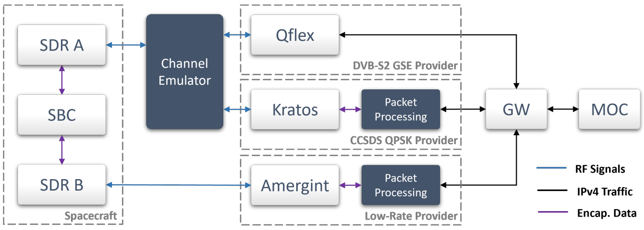

Figure 4: Emulated spacecraft communications testbed used to evaluate the ASE framework.

SDR B emulates a dedicated radio for commanding and low-rate telemetry. This connection is assumed to be an on-demand link always available to the spacecraft for asynchronous transmission of telemetry or reception of commands. For decades, NASA's Tracking and Data Relay Satellite System (TDRSS) has provided this capability with its always-on receivers for certain code division multiple access users [21]. Emerging commercial services (e.g. [22]) plan to offer an equivalent capability. The persistent connection from SDR B to the Amergint modem is not switched through the Propsim. The framework can also support a spacecraft configured with a single radio. In this case, the SDR switches between low-rate and high-rate contacts. The down time between each pair of contacts is $t_{\text{lead}} = 18\text{s}$, primarily due to the calibration routine run after each SDR configuration.

5.1 DVB-S2 GSE

DVB-S2 Generic Stream Encapsulation (GSE) is an adaptation layer protocol designed for efficient encapsulation of IP and other network-layer packets. GSE offers enhanced features compared to other DVB encapsulation schemes such as low overhead, fragmentation, and variable length traffic to enhance performance for techniques such as Adaptive Coding and Modulation (ACM) [23]. The variable length of GSE packets is ideal for transmitting IP packets which commonly vary in length as encountered in modern network traffic. This combined with physical layer fragmentation enables minimal overhead and full utilization of the baseband frame data field [24].

The SDR implementation of DVB-S2 GSE allows for the full two-byte variable length field. During periods of no traffic, GSE idle frames are generated for link stability. For GSE packet sizes exceeding the maximum baseband frame length, physical fragmentation is employed to split GSE frame across multiple baseband frames for later reassembly. To optimize data throughput, GSE High Efficiency Mode is implemented to allow GSE frames to begin filling the baseband data field immediately after the previous packet, rather than starting at the beginning of a new frame. This approach enabled efficient and adaptive transmission of data for the DVB-S2 waveform.

Since the SDR does not itself implement an IPv4 gateway, SBC services are necessary to provide IPv4 packet conversion to the radio's data interface. The SBC's *svcces_dvb* application provides a network interface for traffic to providers which use DVB-S2 protocols. The interface forwards IP datagrams to the SDR over a local Ethernet connection using the vendor's transport protocol.

---

*SDR implementations of these protocols are available for limited release through software.nasa.gov.

5.2 CCSDS IPE

CCSDS defines standards for use in developing data systems for space missions such as the Encapsulation Packet Protocol (EPP) [25], which is implemented in our testbed. Since CCSDS Space Data Link Protocols require data units to have a Packet Version Number (PVN), the EPP provides the capability to transfer data units without an authorized PVN, such as IPv4, over space links. As our testing utilizes IP data, we also adhere to the CCSDS standard for transmitting IP packets over space links [12]. This standard supports the EPP by providing an IP Extension (IPE) that identifies the encapsulated Internet Protocol. The motivation for implementing the EPP is to support IPv4-based applications requiring routable subnets over CCSDS-compliant data links. For IPv4, packets are encapsulated by prepending the packet with an EPP header and IPE (i.e., the output order is EPP header, IPE, IPv4 packet).

On the SBC, the *svcces_epkt* application provides a network interface for the low-rate multiple access links. The interface performs CCSDS encapsulation, forms AOS transfer frames, and forwards them to the SDR. The SDR modulates AOS transfer frame bytes, applies the spacecraft's assigned multiple-access spreading code, and transmits the RF signal. For high-rate CCSDS links, the *svcces_dvb* application is again used to transfer IPv4 frames to the SDR. CCSDS encapsulation is performed in the radio's programmable logic fabric along with QPSK modulation and transmission.

5.3 Network Configuration

The *svcnet* SBC application provides configuration of data links, Linux network interface *qdisc netem* rate control, and IP routing table modification. As in Algorithm 1, the application can be commanded to change routing topology based on an active contact. These commands are sent as JSON objects over a local TCP interface on the SBC. Fig 5 shows the connection between *svcnet*, *svcces_dvb*, and *svcces_epkt* along with their interfaces to the Linux kernel and SDR.

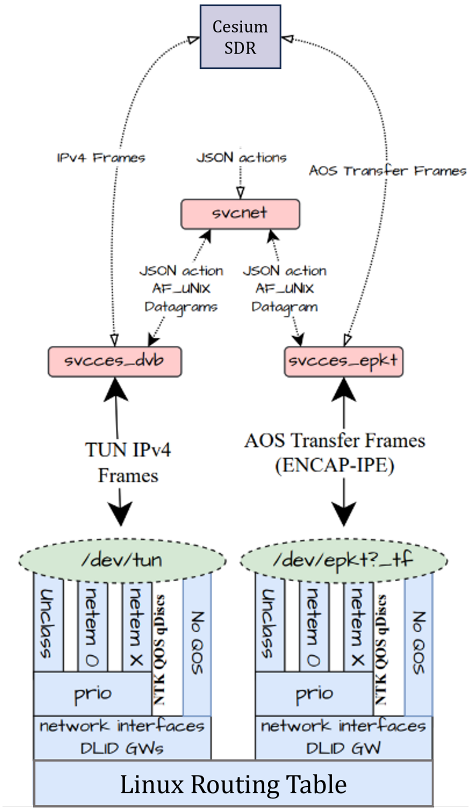

Figure 5: Data flow and configuration of network interfaces used to transport IP between the spacecraft SDR and SBC.

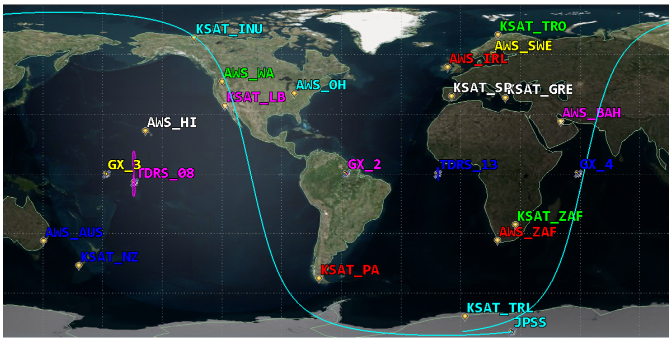

Figure 6: Spacecraft and ground stations used in test scenario.

6 Results

Fig. 6 shows the example scenario developed to evaluate the ASE framework. A science spacecraft in a sun synchronous orbit is assumed, simulated using orbital elements of the first Joint Polar Satellite System (JPSS-1). The spacecraft has periodic connectivity to three geostationary relays, emulated using the orbital elements of Inmarsat Global Xpress (GX) satellites. The scenario also contains two ground station networks emulated using positions of sites operated by Amazon Web Services (AWS) and Kongsberg Satellite Services (KSAT). Additionally, a low-rate link is persistently available through TDRSS' multiple access system.

6.1 Rapid Switching

Fig. 7 shows a brief test with two scheduled contacts through GX and AWS, respectively. When a high-rate scheduled link is not available, data flows over the low-rate link to TDRSS. Data is still able to flow over the low-rate link as SDR A is being configured in advance of each scheduled high-rate contact. At contact start, the framework commands *svcnet* to adjust the routing table. Switchover to the high-rate link is nearly instantaneous.

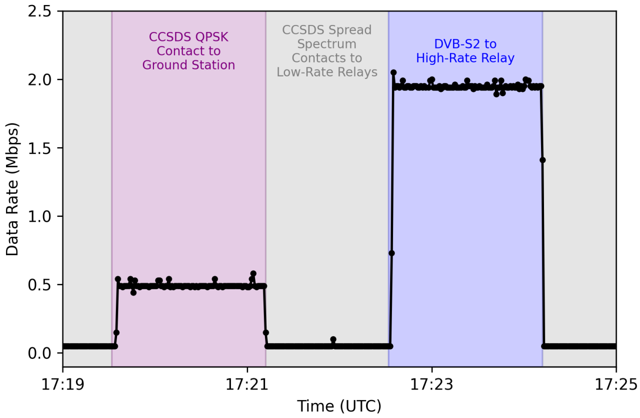

Figure 7: Rapid switching between connections to three providers, each using a different protocol set and data rate.

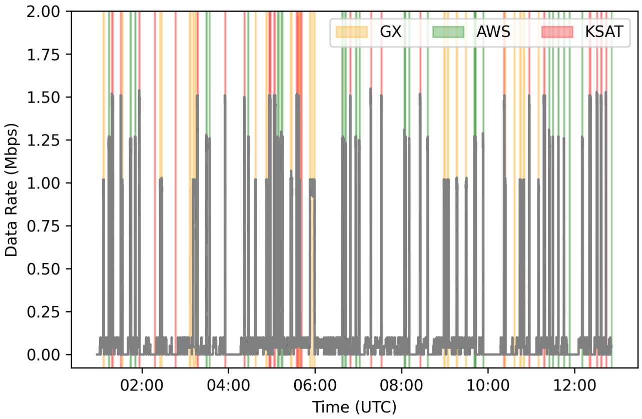

Figure 8: Long run with dozens of contacts scheduled to each of three providers. The ASE framework repeatedly applies the correct configuration for the active link.

6.2 Long Duration Run

In a separate 12 hour test, a total of 69 links scheduled to a variety of providers (Table 2). This number of contacts is unusually high for a real space mission but was meant as a stress test for the framework software. At random intervals parameterized by a Poisson process, data was generated onboard the spacecraft. In response to each data generation event, contacts to the three emulated providers were dynamically scheduled following the process in [7]. The spacecraft's plan was continuously updated as new contacts were added.

The ASE framework started and properly configured end-to-end communication for each scheduled contact. As in the previous test, communication switched to the low-rate TDRSS service between scheduled contacts. DTN application software [26] is used to send data over the intermittently available links. Fig. 8 shows data received at the MOC. A total of 776MB of data was transferred during the test.

The ASE framework provides the unique capability of mapping between DTN-layer contact graph routing and IP-layer routing. The system utilizes dynamic contact plan updates and configures the appropriate IP routing. Contact plan updates and the translation of DTN routes into lower layer routing are two needed capabilities which have not been addressed in any specification to our knowledge.

Provider	Assets	Waveform	Rate	Contacts
TDRSS	3	SS-BPSK	50kbps	-
GX	3	DVB-S2	1.0Mbps	19
AWS	8	QPSK	1.25Mbps	25
KSAT	10	QPSK	1.50Mbps	25

Table 2: Contacts to each emulated service provider successfully configured during the 12-hour run.

7 Conclusion

The goal of a single "multilingual" terminal which can interoperate with the wide variety of communications services has been strongly desired in the mission community. Previous work [2] demonstrated RF-flexible terminals as an enabling technology but heavily relied on pre-scripted actions which are not scalable to many providers. In this work, we presented an extensible framework to automate the process of end-to-end configuration across a variety of different protocols. The proposed automation framework frees the mission operator from repeated step-by-step configuration of links and routes. With assurance that data will reach its destination, missions can focus on execution of their science objectives.

8 Acknowledgments

NASA's Space Communication and Navigation Program, Advanced Communication and Navigation Technology Division provided funding for development and demonstration of this work. We thank Vladimir Linetsky for his assistance in deployment and operation of testbed hardware.

References

[1] Heckler, G., Piasecki, M., Barthelme, N., et al.: 'NASA's efforts to pursue commercial communications services for missions in near space'. In: 28th Ka and Broadband Communications Conference. (Bradford, England, 2023).

[2] 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'. In: Proc. 38th International Communications Satellite Systems Conference (ICSSC). (Arlington, VA, 2021).

[3] 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'. In: Proc. 26th Ka and Broadband Communications Conference. (Arlington, VA, 2021).

[4] Hosseinian, M., Choi, J.P., Chang, S.H., Lee, J.: 'Review of 5g ntn standards development and technical challenges for satellite integration with the 5g network', *IEEE Aerospace and Electronic Systems Magazine*, 2021, 36, (8), pp.22–31

[5] Perkins, C.E.: 'Mobile ip', *IEEE Communications Magazine*, 1997, 35, (5), pp.84–99

[6] Tsunoda, H., Ohta, K., Kato, N., Nemoto, Y.: 'Supporting ip/leo satellite networks by handover-independent ip mobility management', *IEEE Journal on Selected Areas in Communications*, 2004, 22, (2), pp.300–307

[7] Gannon, A., Paulus, S., Gemelas, C., et al.: 'Spacecraft-initiated scheduling of commercial communications services'. In: 39th International Communications Satellite Systems Conference (ICSSC). (Stresa, Italy, 2022).

[8] Pan, J., Zhao, J., Cai, L.: 'Measuring a low-earth-orbit satellite network'. In: International Conference on Communications (ICC). (Denver, CO, 2024).

[9] Hibberd, C.: 'Inmarsat global xpress network - meeting thechallenges of providing a seamless global kabandservice to mobile terminals'. In: 30th AIAA International Communications Satellite System Conference (ICSSC). (Ottawa, Canada, 2012).

[10] 'Advanced orbiting systems, networks and data links: Architectural specification'. (CCSDS Blue Book, 1989). CCSDS 701.0-B-1

[11] Hooke, A.J.: 'CCSDS advanced orbiting systems: international data communications standards for the space station freedom', *IEEE Network*, 1990, 4, (5), pp.13–16

[12] 'IP over space links'. (CCSDS Blue Book, 2012). CCSDS 702.1-B-1

[13] Burleigh, S., Hooke, A., Torgerson, L., et al.: 'Delay-tolerant networking: an approach to interplanetary internet', *IEEE Communications Magazine*, 2003, 41, (6), pp.128–136

[14] Ivancic, W., Stewart, D., Wood, L., et al. 'IPv6 and IPsec tests of a space-based asset, the cisco router in low earth orbit (CLEO)'. (NASA Glenn Research Center, 2008). NASA/TM—2008-215203

[15] Pajevski, M.J.: 'Use of IPsec by manned space missions'. In: IEEE Aerospace conference. (Big Sky, MT, 2009).

[16] 'Space system protection standard'. (NASA, 2022). NASA-STD-1006A

[17] Leppinen, H.: 'Current use of linux in spacecraft flight software', *IEEE Aerospace and Electronic Systems Magazine*, 2017, 32, (10), pp.4–13

[18] 'Schedule-aware bundle routing'. (CCSDS Blue Book, 2019). CCSDS 734.3-B-1

[19] 'CCSDS bundle protocol specification'. (CCSDS Blue Book, 2015). CCSDS 734.2-B-1

[20] Downey, J., Gannon, A., Smith, A., et al.: 'Emulated spacecraft communication testbed for evaluating cognitive networking technology'. In: IEEE Cognitive Communications for Aerospace Applications (CCAA) Workshop. (Virtual, 2023).

[21] Gatlin, T.A., Horne, W.: 'The NASA space network demand access system (DAS)'. In: SpaceOps 2002 Conference. (Houston, TX, 2002).

[22] Pang, T.K., Trachtman, E.: 'Inter-satellite data relay system (IDRS) for LEO satellites using a commercially available GEO satellite system'. In: Proc. Small Satellite Conference. (Logan, UT, 2020).

[23] 'Digital video broadcasting (DVB); generic stream encapsulation (GSE)'. (ETSI, 2014). ETSI TS 102 606-1 V1.2.1

[24] 'Digital video broadcasting (DVB); second generation part 2: DVB-S2 extensions (DVB-S2X)'. (ETSI, 2014). ETSI EN 302 307-2

[25] 'CCSDS encapsulation packet protocol'. (CCSDS Blue Book, 2020). CCSDS 133.1-B-3

[26] Dudukovich, R., Raible, D., Tomko, B., et al.: 'Advances in high-rate delay tolerant networking on-board the international space station'. In: IEEE Space Computing Conference. (Mountain View, CA, 2024).