

An End-to-End GEO Satellite Links Simulation Framework for Cyber Range Applications

Alessandro Santorsola^{1,2}, Daniele Mammone³, Stefano Longari³, Francesco Topputo⁴, and Matteo Merge⁵

¹Cybersecurity Lab, BV TECH S.p.A., 20123 Milan, Italy

²Department of Electrical and Information Engineering, Polytechnic University of Bari, Bari, Italy

⁴Department of Aerospace Science and Technology, Polytechnic University of Milan, Milan, Italy

³Department of Electronics, Information and Bioengineering, Polytechnic University of Milan, Milan, Italy

⁵Italian Space Agency, Rome, Italy

(email: alessandro.santorsola@bvtech.com)

Abstract—Satellite communications represent a cornerstone in modern information and communication systems, enabling a wide range of applications from global positioning and weather monitoring to critical infrastructure management and emergency response. As these systems become increasingly interconnected, their exposure to sophisticated cyber threats grows significantly. The susceptibility of satellite infrastructures to cyber attacks highlights the urgent need for developing, testing, and validating robust security solutions. Cyber ranges represent essential tools for cybersecurity research and testing. This paper presents a novel system-level simulation framework specifically designed for cyber range applications, enabling end-to-end emulation of GEO satellite links from ground stations to space and vice versa. The goal is to provide a realistic and flexible testbed that can support the assessment of security measures, incident response strategies, and the resilience of satellite-ground communications against various attack scenarios.

Index Terms—Cyber Range, Satellite Communication, Cybersecurity, Radio Channel Simulation

I. INTRODUCTION

In modern communication infrastructure, satellite systems are a critical component, enabling several services ranging from telecommunications and navigation to Earth observation and national security [1], [2]. As these systems increasingly adopt Commercial

Off-The-Shelf (COTS) technologies [3], their attack surface expands, exposing them to a series of sophisticated cyber-physical threats [4], [5]. In fact, the integration of heterogeneous systems, including both terrestrial and non-terrestrial networks, amplifies the satellite-based network's attack surface. Therefore, the development of robust cybersecurity measures to safeguard interconnected infrastructures has become essential to ensure long-term resilience. Satellite communications and systems security are gaining increasing attention within the scientific community. Recent research highlights growing concerns over the potential exploitation of vulnerabilities in satellite-based data services, including adversarial manipulation of satellite observations in applications such as weather forecasting [6]. This trend reflects a broader shift in focus, with researchers investigating threat modeling, anomaly detection, secure communication protocols, and resilience strategies for spaceborne systems [7]–[9]. Thus, the convergence of cybersecurity and space technologies creates a new interdisciplinary frontier that aims to protect the integrity and reliability of satellite-enabled infrastructures.

Cyber ranges are becoming essential tools for advancing cybersecurity research and solutions development [10] [11]. Cyber Ranges are controlled and high-fidelity simulation environments that enable the testing of realistic attack scenarios, the validation of defensive strategies, and the training of operators under adversarial conditions [12]. In particular, cyber ranges tailored to space systems allow the emulation of satellite-to-ground communication and supply chain compromises that are crucial for assessing the resilience of both legacy and next-generation space infrastructures [13]. The European Space Agency (ESA) space cyber range provides a virtual-physical environment for communication-level threat simulation and operator training [14].

This work was developed under the Satellite Operations Cyber RAnge for Testing and Evaluation (SOCRATE) Project. It was managed and funded by Agenzia Spaziale Italiana (ASI) through Mid-Term Research and Development projects “Giornate della Ricerca Accademica Spaziale” (Research Day) ASI 2020, within the Thematic Tables “Scientific Instrumentation, Cybersecurity, and Advanced Materials”. Thematic Area for “Design and development of simulation platform of space systems to execute cyber attack/defense scenarios as well as terrorist attack scenarios, natural disasters or space weather events to help operators to analyze the effectiveness of human and technical responses”.

However, the lack of support for end-to-end testing across space and ground segments represents a limitation. The NASA Operational Simulator for Small Satellites (NOS3) supports flight software development and component-level testing in small satellite missions, but lacks features for integrated cybersecurity validation, threat modeling, and standardized protocol emulation, such as Consultative Committee for Space Data Systems (CCSDS) protocols [15]. The Unified Cybersecurity Testing Lab introduces modular test beds, including Software-Defined-Radios (SDRs) to emulate multi-domain protocols [16]. The OpenSatRange platform proposed in [17] is focused on scalable satellite communication emulation and configurable scenarios within a cloud-based environment. However, it does not currently integrate CCSDS protocols testing or structured operator training frameworks. Finally, authors in [13] propose attack scenarios for OpenSatRange targeting satellite system vulnerabilities.

This work presents a system-level simulation framework for the emulation of GEO satellite communication links within cyber ranges. The architecture supports high-fidelity, end-to-end modeling from ground station application data to the satellite and back, featuring full implementation of the CCSDS and Space Link Extension (SLE) protocol stacks, bit-stream generation, and Telecommands and Telemetry (TC/TM) waveform handling. The framework enables the creation of realistic and customizable scenarios by combining protocol-level emulation with detailed physical channel modeling. In summary, the key features of the proposed framework include:

- Full emulation of CCSDS and SLE protocol stacks;
- Flexible modeling of satellite channels, including environmental and atmospheric effects;
- Generation and handling of CCSDS TC/TM waveforms.

The framework is designed for extensibility and fine-grained control down to the bit level, thanks to a queue-based communication overlay layer. It further incorporates real-time weather-based attenuation, channel interference estimation, and multi-band signal combination for spectral overlap modeling. Space weather effects, such as solar flare-induced interference, are also considered. Therefore, the proposed framework enables operational exercises requiring realistic physics simulation, such as jamming detection in Ka-band, link-margin assessment under ITU-R P.618 rain/scintillation, co-channel interference, and recovery from solar conjunction events.

Compared to the ESA Cyber Range [14] and the Unified Cybersecurity Testing Lab [16], this work

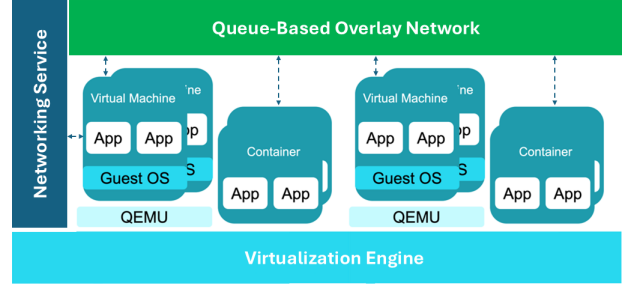


Fig. 1: General Architecture of the queue-based modules communication.

integrates complete CCSDS and SLE protocol stacks with ITU-compliant RF channel modeling, enabling end-to-end emulation rather than packet-level replay. In contrast to NOS3 [15], which focuses on flight software and spacecraft dynamics, our framework focuses on transmission emulation and protocol-level evaluation. OpenSatRange [17], on the other hand, provides RF loopback and interference tests but does not include waveform-level integration with CCSDS/SLE. Accurate RF modeling is essential for simulating scenarios that cannot be represented by simple delay or packet-loss emulation. These include wideband and narrowband jamming, adjacent-channel interference, cross-polarization and pointing losses, and space weather events such as solar flares. Such effects directly impact link performance, and, consequently, determine the resilience of CCSDS/SLE protocol stacks during cyber exercises. The framework therefore enables training and assessment under conditions where physical-layer impairments are a critical part of the security challenge.

The remainder of this paper is organized as follows: Section II details the system architecture. Sections III and IV report the reference deployment, settings, and validation results within a cyber range context. Finally, Section V concludes the paper and discusses future extensions.

II. FRAMEWORK ARCHITECTURE

Cyber ranges rely on virtualization platforms to create isolated, and reproducible environments. Common solutions like VMware ESXi [18], Microsoft Hyper-V [19], KVM/QEMU [20], and Proxmox VE [21] support the deployment of virtual machines (VMs) and containers hosting critical components. Hypervisors provide advanced networking features such as virtual bridges, VLANs, and software-defined networks (SDN), enabling fine-grained control over bandwidth, latency, routing, and isolation. Network isolation is essential in cyber range design, as executing attacks on non-isolated networks can lead

to unintended traffic propagation or legal issues. Isolated setups, on the other hand, ensure safety and allow full observability through monitoring probes and traffic inspection tools.

A. General Architecture

1) *Queue-Based Modules Architecture*: Figure 1 depicts the proposed queue-based communication architecture, designed for virtualized environments supporting both VMs and containers. The virtualization engine abstracts hardware resources and hosts heterogeneous instances representing functional components such as ground stations, terminals, or mission-critical applications. Each instance includes a custom guest OS and an application layer. All the components communicate via a queue-based overlay network that models event-driven interactions, i.e., signal reception, featuring fine-grained control over packet flow, latency, and delay. A central queue service manages asynchronous message exchange over the underlying TCP/IP stack, leveraging the hypervisor's network services. This architecture supports several simulation scenarios, including low-level protocol manipulation and dynamic topology reconfiguration, while remaining hypervisor-independent.

2) *Queue-Based Module Design*: Figure 3 reports a simplified conceptual view of the internal organization of a single application, e.g., ground station, Uplink/Downlink, etc., within the overall framework. The design follows a block-based approach for service logic, communication handling, persistent data management, and API exposure. The modular design ensures the reproducibility of the system, enabling individual components to be developed, tested, and independently deployed.

3) *Message Structure*: Figure 2 shows the message structure used within the queue-based overlay communication network. The message is composed of two main branches: (i) Uplink/Downlink, and (ii) LinkInfo. The Uplink/Downlink branch contains raw bitstreams, i.e., the input and the output binary packets, as well as a set of communication performance statistics, including channel capacity, signal-to-noise ratio (SNR), bit error rate (BER), and transmission errors. The LinkInfo branch encapsulates satellite link contextual metadata, such as the timestamp and all the detailed layout information of the communication link, including ground station parameters (e.g., elevation angle, position, height) and satellite attributes (e.g., latitude, longitude, altitude).

B. Uplink/Downlink Channels

The Uplink and Downlink channels have been designed according to the CCSDS specification [22]–

[26]. We built the transceiver modules based on the examples and functions provided by the MATLAB Satellite Communication Toolbox [27], [28]. The Satellite Communication Toolbox implements all the necessary functions, including waveforms, receiver processing, etc. In general, both uplink and downlink are composed of two main modules: (i) Radio Frequency (RF) CCSDS Frontend, and (ii) Radio Propagation model, respectively.

1) *Radio Frequency CCSDS Frontend*: Figure 4 (a) reports the RF frontend architecture that generates a complex baseband CCSDS TC/TM waveform from the incoming packets, i.e., the bitstring. The incoming bitstream is arranged into a CCSDS transfer frame of a maximum length of 800 octets. The resulting signal will be amplified up to the transmission power, and, finally, an additive white Gaussian noise (AWGN), evaluated according to the specific earth-space layout, will be added to the signal. The lower process chain shows the synchronization, demodulation, and decoding of the received signal to get the outPacket.

2) *Radio Propagation Model*: Figure 4 (b) reports the radio propagation blocks that are used to evaluate the satellite link budget, taking into account all the relevant information of earth-space layout, path loss, atmospheric losses, and interference. The effective isotropic radiated power (EIRP) is calculated in decibel milliwatt as:

$$\text{EIRP} = P_{tx} - L_f - L_{tx} + G_{tx} \quad (1)$$

where P_{tx} is the transmitting signal power, i.e., post-amplified, L_f the feeder loss, L_{tx} other transmission losses, and G_{tx} the antenna gain.

The free-space path loss (FSPL) is computed using the Friis equation:

$$L_{fs} = 20 \log_{10} \left(\frac{4\pi d}{\lambda} \right) \quad (2)$$

where d represents the distance between the ground station and the satellite in meters, and λ is the wavelength. Atmospheric attenuation L_{atm} and polarization mismatch L_{pol} losses are subtracted from EIRP to obtain the isotropic received power:

$$P_{rx} = \text{EIRP} - L_{fs} - L_{atm} - L_{pol} - L_{pointing} \quad (3)$$

The $L_{pointing}$ factor takes into account possible mispointing losses, e.g., 1 dB.

The carrier-to-noise density ratio is derived as follow:

$$\frac{C}{N_0} = P_{rx} + \frac{G}{T} - 10 \log_{10}(k) - L_{rf} \quad (4)$$

where G represents the antenna gain in dBi, and T is the equivalent noise temperature of the receiving

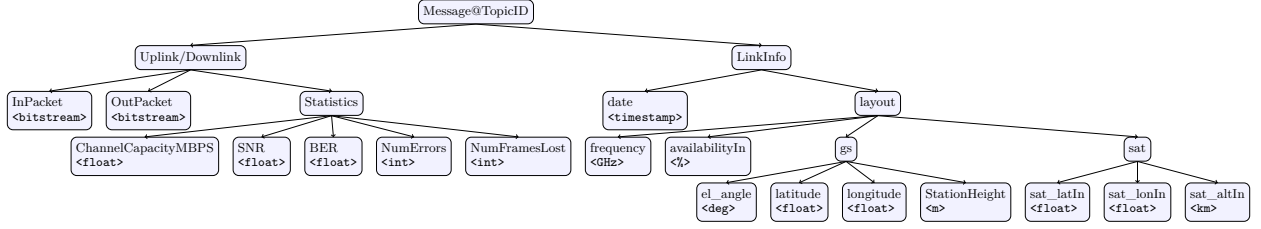


Fig. 2: Hierarchical structure of a message over the queue-based overlay network.

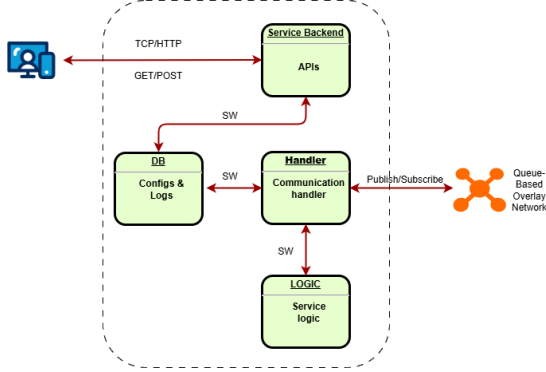


Fig. 3: Base Architecture of a queue-based module.

system in Kelvin. Therefore, the ratio G/T is the receiver figure of merit. Finally, k is Boltzmann's constant, and L_{rf} are receiver losses.

The energy-per-bit-to-noise density ratio is computed as:

$$\frac{E_b}{N_0} = \frac{C}{N_0} - 10 \log_{10}(R_b) \quad (5)$$

where R_b is the transmitter bit rate.

$$\text{Margin} = \frac{E_b}{N_0} - \left(\frac{E_b}{N_0} \right)_{\text{req}} - L_{\text{impl}} \quad (6)$$

where L_{impl} takes into account additional system implementation losses.

Finally, the SNR is evaluated as:

$$\text{SNR} = \frac{E_b}{N_0} - L_{\text{impl}} + r_{\text{dB}} + M_{\text{dB}} - \text{sps}_{\text{dB}} \quad (7)$$

where r is the coding rate, M the modulation order, and sps the samples per symbol.

a) Atmospheric Attenuation: The atmospheric attenuations are modeled by combining the ITU-R P.618 propagation loss model [29] with real-world meteorological data retrieved from *Meteostat* [30]. The framework uses the MATLAB p618 Propagation Losses, which is the base implementation of the ITU-R P.618 recommendation [31]. This utility computes the total propagation losses, including rain attenuation, cloud/fog, and tropospheric scintillation contributions, based on frequency, elevation angle, and

atmospheric parameters, i.e., temperature, pressure, relative humidity, and rainfall amount recorded over a given period, used to estimate the rain rate. In the proposed extension, this model is dynamically driven by real-time or historical weather observations. Specifically, the system queries the *Meteostat* service using the geographic coordinates of the ground terminal, selecting the closest meteorological station. The meteorological data corresponds to horizontal surface measurements taken from the nearest weather station. Although the satellite signal traverses the atmosphere vertically, surface-level meteorological parameters are used for vertical profiles within the ITU-R P.618 model [29]. This simplification is acceptable in non-critical scenarios, such as cyber range training, where repeatability and integration are prioritized over absolute accuracy. The retrieved parameters affect the following atmospheric loss components:

- Rain attenuation L_{rain} is computed as a function of rain rate and frequency,
- Gaseous attenuation L_{gas} depends on temperature, pressure, and humidity,
- Cloud and fog loss L_{fog} are affected by liquid water content and cloud thickness.

The final atmospheric loss L_{atm} is then calculated as the sum of all components:

$$L_{\text{atm}} = L_{\text{rain}} + L_{\text{gas}} + L_{\text{fog}} + L_{\text{sci}} \quad (8)$$

where L_{sci} factor takes into account the tropospheric scintillation loss.

3) Interference Modeling:

a) Solar Flares: The proposed model integrates an electromagnetic (EM) interference module that estimates the power contribution of the Sun as a noise source during periods of solar conjunction or flare activity. The implementation is aligned with ITU-R S.1525 [32], which provides analytical expressions for solar radio emission contributions to the noise temperature and $\frac{C}{N_0}$ degradation [32]. The proposed model considers the Sun aligned behind the satellite from the Earth's perspective. Therefore, it contributes to high-intensity radio emissions that

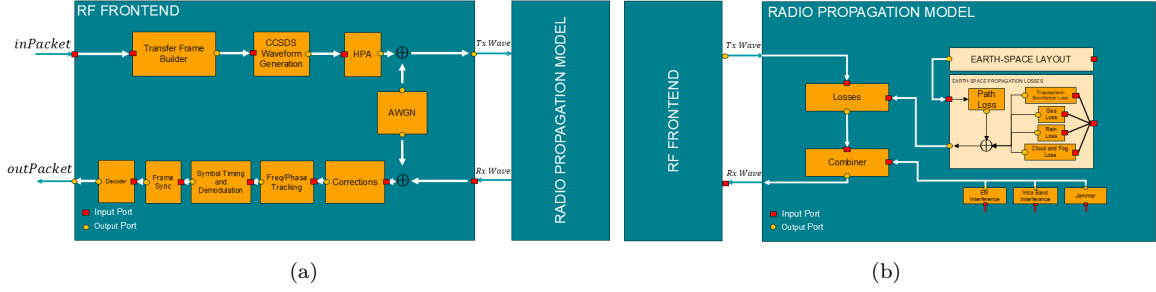


Fig. 4: Schematic architecture of the radio frontend Tx/Rx frame processing steps (a), and of the radio propagation model modules (b).

degrade the SNR. According to [32], the interference is based on solar radio flux density and antenna characteristics. For the sake of clarity, the simulation does not dynamically compute orbital positions to determine conjunction events. Instead, the phenomenon is abstracted as a boolean condition, i.e., active/inactive. Future developments will integrate realistic orbital dynamics to model conjunctions based on actual Sun/Earth/Satellite positions.

The solar power P_{sun} received at the antenna is computed as:

$$P_{\text{sun}} = S_{\text{sun}} \cdot A_{\text{eff}} \cdot B \quad (9)$$

where:

- S_{sun} is the solar power spectral density in $\text{W}/\text{m}^2/\text{Hz}$,
- $A_{\text{eff}} = \eta \cdot \pi \left(\frac{D}{2}\right)^2$ is the effective antenna area expressed in m^2 , with diameter D and efficiency η ,
- B is the system bandwidth in Hz.

The resulting solar noise power is expressed in dBW as:

$$P_{\text{sun,dBW}} = 10 \log_{10}(P_{\text{sun}}) \quad (10)$$

This value can be directly introduced into the interference budget of the link as an additive noise term when computing C/N_0 or SINR, particularly under worst-case solar activity. Therefore, the SINR can be evaluated as:

$$\text{SINR} = \frac{P_{rx}}{P_{\text{noise}} + P_{\text{int}}} \quad (11)$$

where P_{rx} represents the legitimate received signal power, P_{noise} takes into account the noise power, and P_{int} takes into account the interference power, which may include contributions from solar events or malicious RF emitters (if any). The interference impact is especially relevant in GEO satellite links

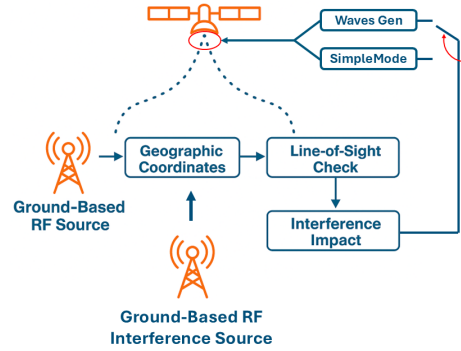


Fig. 5: Basic architecture of the interference management handling

during solar transit periods, where antenna alignment toward the Sun may coincide with the satellite direction. The resulting effective energy-per-bit to noise-plus-interference ratio (E_b/N) is computed starting from the SINR by taking into account the modulation order, coding rate, and oversampling factor as follows:

$$\frac{E_b}{N} = \text{SINR} - r_{\text{dB}} - M_{\text{dB}} + \text{sps}_{\text{dB}} \quad (12)$$

b) Intentional and unintentional interference:

The framework incorporates both intentional (e.g., jamming) and unintentional (e.g., spectral overlap) interference sources through a waveform-level superposition approach. This approach allows the simulation of overlapping multiple signals arriving at the receiver, such as co-channel emitters, broadband jammers, or adjacent-band interferers.

The interference is modeled as one or more distinct baseband signals, modulated with potentially different carrier frequencies, bandwidths, and power levels. Clearly, the interference signal is treated as undergoing the same transmission process previously described. Therefore, the framework uses the geographic coordinates of the interfering transmitter

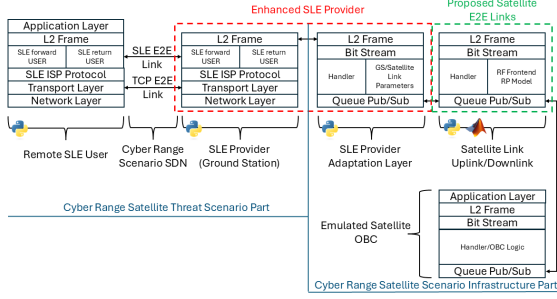


Fig. 6: Schematic of the SLE Remote User(s), SLE Provider, and Communication links stacks integration.

to assess whether the source has a valid line-of-sight to the satellite. This ensures that only realistic interference contributions are considered, accounting for antenna elevation angle, satellite position, and propagation conditions. Figure 5 reports the basic flow of the aforementioned interference handling in the case of two ground stations. The received signals are then combined into a single composite waveform:

$$y_{rx}(t) = y_{sig}(t) + \sum_{i=1}^N y_{int,i}(t) \quad (13)$$

where $y_{sig}(t)$ is the useful transmitted signal and $y_{int,i}(t)$ are the interference components. The final waveform $y_{rx}(t)$ represents the real-time input received by the satellite's RF frontend. Therefore, the following effect can be simulated: (i) narrowband or wideband jamming attacks, (ii) in-band uncoordinated transmissions, (iii) frequency-adjacent interference, usually due to misconfigured emitters, and (iv) inter-satellite link contamination in dense constellations.

The framework supports scalable operational modes based on available resources, starting from lightweight emulation for constrained environments to full waveform-level signal processing when sufficient computational capacity is available. The lightweight configuration models channel degradation analytically, without waveform generation, by computing the expected SNR or SINR using the radio propagation model. The resulting degradation is then directly applied to the bitstream by injecting bit errors at a rate consistent with the expected BER.

C. Space Link Extension (SLE) and CCSDS Protocols Integration

Figure 6 illustrates the proposed integration of the SLE and CCSDS protocol stack within the virtualized cyber range environment. The architecture is

divided into two main parts: (i) the satellite threat scenario, and (b) the satellite scenario infrastructure. In a nutshell, the first part emulates the SLE User(s) and Ground Station Provider(s) roles, while the second part simulates the satellite wireless links modeled according to the above-mentioned queue-based architecture.

The SLE User and Provider modules are interconnected via a TCP/IP stack [33] managed by the virtualization platform. The SLE Provider has been developed by using the open-source `visionspacetec/sle-provider` [34]. For the purpose of this work, the SLE Provider reported in [34] has been extended to interface with a queue-based infrastructure, enabling the decoupled transmission and reception of CCSDS Level-2 Frames through publish/subscribe paradigm. The bitstream will be generated from each transfer frame and passed across queues. Within the satellite scenario infrastructure part, the uplink/downlink chains are connected to a queue endpoint that receives bitstreams from the SLE-based system and vice versa. Figure 6 introduces a conceptual module representing the satellite's OBC (On-Board Computer), responsible for further processing of received telemetry and telecommand data. For the sake of clarity, the satellite OBC architecture, processing logic, and validation are out of the scope of this paper and will be addressed in future works. The aforementioned architecture is written in Python and MATLAB. The former is used for protocol and logic handling (SLE, CCSDS, queue processing), while the latter is employed for link modeling and signal-level analysis. The SLE User has been implemented by using the `librecube/python-sle` library [35]. Finally, the `spacepackets` Python library [36] has been used to generate the Level-2 CCSDS Transfer Frames. Further extensions have been added to support binary frame generation, frame sequence management, and user data field insertion, such as the CCSDS Function packets, adhering to the CCSDS 132.0-B and 133.0-B recommendations. The integration between the modules has been validated through an automated Git-based CI/CD pipeline [37], which executes unit and interface tests across all key components. The set of validation procedures are summarized in the Appendix.

III. REFERENCE DEPLOYMENT AND VALIDATION SETTINGS

Figure 7 reports the reference deployment topology implemented during system testing and validation. The diagram shows a segmented network embedding three principal zones: (i) the Command & Control Network (hosting the SLE User), (ii) the Ground

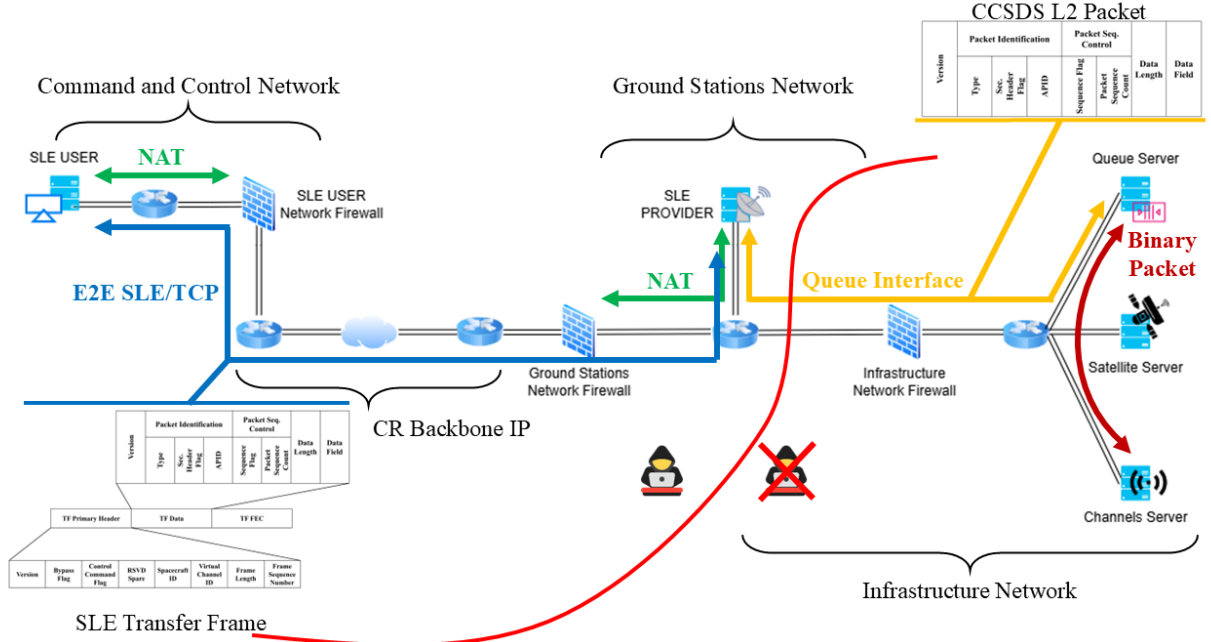


Fig. 7: Reference deployment topology for SLE, queue and infrastructure services. Firewalls and NAT simulate real-world access policies. The red boundary represents the attack-exclusion zone protecting the internal infrastructure network. The blue arrow highlights the end-to-end SLE over TCP data flow, while the yellow arrow highlights the flow of CCSDS Layer 2 frames across the system architecture. The SLE PDU and CCSDS frame structures are embedded in the figure, illustrating their encapsulation and flow along the communication path. The figure also includes the Uplink/Downlink server interfaces responsible for satellite channel simulation, and the satellite server. The brown arrow highlights the bitstream communication between the Queue and the Uplink/Downlink server.

Stations Network (hosting the SLE Provider), and (iii) the Infrastructure Network (hosting internal components such as the queue and channel servers). Each zone is firewall-protected to simulate network policies and real-world segmentation. The NAT (Network Address Translation) service is used to expose only selected services, such as the SLE Provider, while keeping internal services (e.g., queues, satellite link models, and satellite models) hidden and isolated. The red boundary defines the trust boundary, i.e., the Infrastructure Network, that is intentionally made unreachable from external attackers within the cyber range scenario, ensuring the integrity of back-end components. The above-mentioned deployment uses a virtualization server running Proxmox VE 8.2.4, equipped with a 112-core Intel Xeon Platinum 8176 (2.10 GHz) CPU and 1 TB of RAM. All services were containerized via Docker to ensure modularity and deployment reproducibility.

The selected deployment topology can be further expanded to emulate a complex ground station net-

work and other departments' network segments, enabling the evaluation of lateral movements, internal threats, and supply chain compromises. Figure 7 also illustrates the SLE PDU and CCSDS frame structures, their encapsulation, and flow along the communication path. The blue arrow highlights the end-to-end SLE over TCP data flow, while the yellow arrow highlights the transfer of CCSDS Layer 2 frames across the system architecture. Finally, the brown arrow highlights the bitstream communication between the Queue and the Uplink/Downlink server.

Table I summarizes the reference parameters used within the system-level validation phases. All the validation results are based on a simplified but realistic configuration involving a single ground station and a geostationary satellite (GEO). The ground station is geographically located at a latitude of 40.53° , and a longitude of 17.43° , while the satellite is positioned in a nominal GEO slot with slight inclination (Latitude: 3.06° , Longitude: 0.11°) at an orbital altitude of approximately 38,224 km. Transmitter and receiver

TABLE I: Reference Configurations

Parameter	Value	Unit / Note
<i>Ground Station (GS) Configuration</i>		
Latitude	40.53	degrees
Longitude	17.43	degrees
Altitude	120	meters
<i>Parasite GS Configuration</i>		
Latitude	40.63	degrees
Longitude	17.94	degrees
Altitude	120	meters
<i>Satellite (SAT) Configuration</i>		
Latitude	3.06	degrees
Longitude	0.11	degrees
Altitude	38224	km
<i>Transmitter (GS)</i>		
HPA Output Power	23	dBW
Output Back-Off (OBO)	6	dB
Feeder Loss	2	dB
Other Losses	3	dB
Antenna Gain	46.5	dBi
<i>Receiver (SAT)</i>		
Interference Loss	2	dB
G/T (Gain over Temp)	25	dB/K
Feeder Loss	1	dB
Other Losses	1	dB
<i>Link Properties</i>		
Carrier Frequency	30	GHz
Bandwidth	36	MHz
Bit Rate	10	Mbps
Required E_b/N_0	10	dB
Availability	99.9	%
Implementation Loss	2	dB
Polarization Mismatch	45	degrees
Antenna Mispointing Loss	1	dB
Radome Loss	1	dB
<i>Solar Interference Power Estimation</i>		
Antenna diameter	1.2	m
Antenna efficiency	60	%
Solar flux density	1.1×10^{-21}	$W/m^2/Hz$

hardware parameters are set to typical values for operational Ka-band systems. The physical layer assumptions include a link bandwidth of 36 MHz, a target bitrate of 10 Mbps, and a required E_b/N_0 of 10 dB. Atmospheric attenuation, implementation losses, and polarization mismatch are also incorporated in the model to reflect realistic link impairments. The assumption of a single GS–GEO configuration provides a controlled, traceable baseline for quantifying the behavior of the proposed architecture under both nominal and adverse conditions. Extensions to LEO/MEO constellations and heterogeneous ground stations are part of planned future work.

IV. PHYSICAL MODEL VALIDATION

Figure 8 reports a comprehensive evaluation of link performance under different environmental and system-level conditions, using a fixed reference layout and key parameters. Figure 8 (a) shows how the link quality, expressed as energy-per-bit to noise power spectral density (E_b/N_0), deteriorates as the frequency increases from 20 to 40 GHz. This behavior

is expected due to the higher free-space path loss (FSPL) and the increased atmospheric attenuation at higher frequencies. In addition, a disruptive drop is visible at 30 GHz, where a jammer is introduced, i.e., Pirate GS. The jammer’s contribution is modeled explicitly, and its impact is visualized through a distinct marker, validating the framework’s interference injection capabilities. Figure 8 (b) illustrates the received solar power (in dBW) due to solar radio flux as a function of the receiving antenna diameter. The interference increases with the aperture area, confirming that large antennas, while improving gain, also capture more broadband noise from other sources, i.e., the Sun. In Figure 8 (c), the link performance is further evaluated in the presence of solar flare events with three different power spectral densities, i.e., 1.1×10^{-21} , $\times 10^{-21}$, and $1 \times 10^{-20} W/m^2/Hz$. These quantities correspond to quiet, moderate, and active solar conditions, respectively. As expected, the E_b/N degrades progressively as the solar intensity increases and with larger antenna apertures, demonstrating the framework’s ability to account for time-varying space weather phenomena. Figure 8 (d) examines the link margin, i.e., the difference between actual and required E_b/N_0 as a function of transmission bitrate. Higher bitrates reduce the energy per bit, thus reducing the margin. The curve confirms that under nominal channel conditions, the link is sustainable up to mid-range bitrates, after which the margin approaches zero, potentially leading to outages. Figure 8 (e) presents a dual-axis plot that links atmospheric rain rate to both attenuation and E_b/N_0 . As expected, the increasing rain rate leads to a rise in total attenuation, which directly contributes to a linear degradation in E_b/N_0 . The adoption of the ITU-R propagation models, coupled with meteorological inputs, confirms the framework’s support for realistic weather-aware performance estimation. Figure 8 (f) shows the power spectral density (PSD) of three signals involved within the model validation: (i) the legitimate QPSK signal, (ii) the interfering GMSK signal, and (iii) their combination. The legitimate QPSK signal (orange) is centered in frequency and exhibits a spectral shape consistent with pulse-shaped modulation using a raised cosine filter. The interfering GMSK signal (yellow) is offset in frequency by 30 kHz and shows the typical compact spectral footprint of GMSK modulation. When the two signals are combined (blue curve), their spectral overlap becomes evident, leading to non-negligible in-band interference. The model validation described above focuses on internal consistency checks and qualitative alignment with ITU-R propagation models. A quantitative comparison with

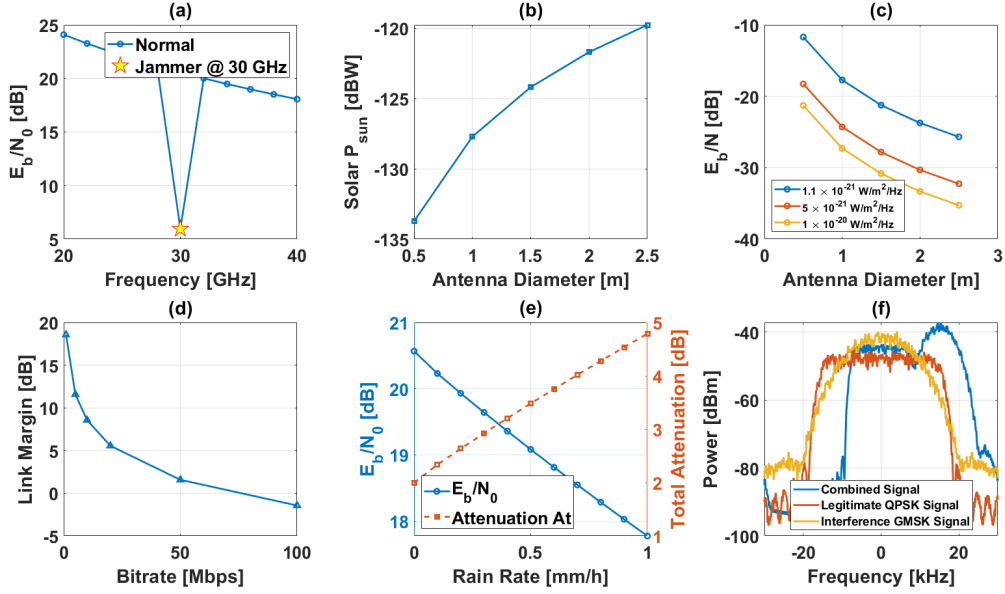


Fig. 8: Satellite Link Models Validation Figures of Merit. E_b/N_0 as a function of transmission frequency (a), Solar Interference Power as a function of Antenna Diameter (b), E_b/N degradation in different Solar Flare conditions (c), Satellite Link Margin as a function of the bitrate E_b/N_0 (d), Total Attenuation as a function of rain rate (e), and Power spectral density of a legitimate QPSK signal and a GMSK interference signal, along with their combined waveform (f).

measured satellite link data or third-party simulators is planned as future work, as it would require access to datasets beyond the current institutional scope.

V. CONCLUSION

This work presented a modular simulation framework designed for end-to-end satellite communication links, specifically tailored for cyber range and emulation environments. The architecture integrates CCSDS/SLE protocol modeling, ITU-compliant propagation effects, and advanced interference simulation, including jamming, spectral overlap, and solar flare scenarios. Queue-based communication enables decoupled interaction between ground stations and virtualized satellite services. Functional validation was carried out using automated GitLab pipelines to test CCSDS packet structures and telemetry handling, confirming the significant impact of both environmental and intentional disturbances on link-level metrics such as E_b/N_0 , link margin, and BER. This integration demonstrates that realistic physical-layer behavior can directly influence cyber-defense response strategies, enabling trainees to observe the operational effects of attacks and countermeasures in real-time rather than only theoretical metrics. Future developments will focus on increasing the fidelity and applicability of the framework by

supporting a wider range of orbital configurations, such as LEO, MEO, and a more accurate simulation of interference patterns. In addition, future evaluations will involve operator teams in realistic cyber-range exercises to assess the framework effectiveness in training and incident-response testing, as well as quantitative benchmarking and large-scale scalability tests.

APPENDIX

This appendix provides additional implementation and validation details related to protocols handling, memory usage, and timing characteristics of the framework components.

The CCSDS and SLE packet handling components were validated through automated unit tests integrated into a GitLab CI/CD pipeline [37], which provides consistency across software revisions and deployments. The test suite verifies encoding, decoding, and integrity checks in full compliance with CCSDS specifications. Specifically, the CCSDS module has been tested for frame construction and parsing, including Test, Function Execution, and Housekeeping packet types. The validation process of the SLE telemetry interface includes socket initialization, encoding of Telemetry Transfer Frames (TTF), and robust handling of edge cases such as invalid service

TABLE II: Summary of CCSDS and SLE Unit Tests

Component	Test Description	Method
Test Packet	Generate CCSDS Test packet and validate binary	<code>test_FunctionTest</code>
	Unpack Test packet from bytes	<code>test_FunctionTestUnpack</code>
	Unpack Test packet from binary	<code>test_FunctionTestUnpackBin</code>
FunctionExecution	Generate unencrypted execution packet	<code>test_FunctionEx</code>
	Unpack from bytes	<code>test_FunctionExUnpack</code>
	Unpack from binary string	<code>test_FunctionExUnpackBin</code>
	Unpack payload fields	<code>test_FunctionExUnpackPayload</code>
Housekeeping	Generate packet with data	<code>test_Housekeeping</code>
	Unpack from bytes	<code>test_HousekeepingUnpack</code>
	Unpack from binary string	<code>test_HousekeepingBin</code>
SLE TM Endpoint	Socket and thread initialization	<code>test_init_creates</code>
	Trigger frame send on queue threshold	<code>test_send_framesthreshold</code>
	Forward valid CCSDS packet to queue	<code>test_forward_buffers_packet</code>
	Handle unsupported service types	<code>test_forward_unrecognizedsrv</code>

TABLE III: Memory Usage and Processing Delay for Fullwave (FW) and Simple Modes

Service	Memory Usage		Delay	
	FW	Simple	FW	Simple
Uplink	3 GB	2 GB	< 1.5 s	< 1 s
Downlink	3 GB	2 GB	< 1.5 s	< 1 s
SLE Provider	180 MB	180 MB	-	-
SLE User	37 MB	37 MB	-	-
MQTT Broker	2 GB	2 GB	-	-

identifiers and boundary conditions during packet forwarding. A summary of the test coverage is provided in Table II.

Table III reports the memory usage of each component in both simple and full waveform processing modes, along with the associated processing delays introduced by the channel emulation modules. These delays are measured as the average time required to process a single data burst. In simple mode, processing latency remains well below one second, while full waveform simulation may reach up to 1.5 seconds. These values still meet real-time requirements for most operational use cases, such as interactive control link emulation or telemetry stream testing. Specifically, the uplink and downlink channel models require approximately 2–3 GB of memory, depending on the deployment configuration. Other services, such as the SLE Provider, SLE User, and MQTT-based queue broker, require moderate memory resources and do not contribute significantly to latency. The queue-based architecture inherently supports parallel processing of multiple satellite links. While this work reports single-link performance metrics, multi-link scalability tests are part of ongoing development and will be reported in future extensions.

It is worth pointing out that the channel modules introduce a baseline propagation delay representative of the physical transmission time of a satellite link evaluated as follows:

$$T_p = \frac{d}{c} \quad (14)$$

where T_p is the propagation delay, d is the distance between the ground station and the satellite, and c is the speed of light in vacuum ($\sim 3 \times 10^8$ m/s). The distance d is computed by converting the geodetic coordinates of the satellite and ground station into Earth-Centered Earth-Fixed (ECEF) coordinates using the WGS-84 ellipsoid model [38]. The relative displacement vector is then projected into the local North-East-Down (NED) frame centered at the ground station. The resulting Euclidean norm provides the line-of-sight distance used to evaluate T_p .

REFERENCES

- [1] J. Li, Y. Pei, S. Zhao, R. Xiao, X. Sang, and C. Zhang, “A review of remote sensing for environmental monitoring in china,” *Remote Sensing*, vol. 12, no. 7, p. 1130, 2020.
- [2] N. Joubert, T. G. Reid, and F. Noble, “Developments in modern gnss and its impact on autonomous vehicle architectures,” in *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2020, pp. 2029–2036.
- [3] M. R. Maheshwarappa, M. Bowyer, and C. P. Bridges, “Software defined radio (sdr) architecture to support multi-satellite communications,” in *2015 IEEE Aerospace Conference*. IEEE, 2015, pp. 1–10.
- [4] L. Zhang, Y. Du, and A. Li, “Rapid cascading risk assessment and vulnerable satellite identification schemes for leo satellite networks,” *Reliability Engineering & System Safety*, vol. 256, p. 110699, 2025.
- [5] J. S. Bardin, “Satellite cyber attack search and destroy,” in *Computer and Information Security Handbook*. Elsevier, 2025, pp. 1561–1580.
- [6] E. Imgrund, T. Eisenhofer, and K. Rieck, “Adversarial observations in weather forecasting,” *arXiv preprint arXiv:2504.15942*, 2025.
- [7] J. L. Fleming, “Securing space systems as critical infrastructure: A transformative approach to cybersecurity and risk mitigation,” Ph.D. dissertation, Capitol Technology University, 2025.
- [8] B. William, E. Frank, and J. Kate, “Cybersecurity in orbit: Safeguarding space systems against emerging threats,” 2025.
- [9] B. William, A. Ibrahim, and J. Kate, “Securing the final frontier: Cybersecurity strategies for satellites and space communications,” 2025.

- [10] S. Khan, A. Volpatto, G. Kalra, J. Esteban, T. Pescanoce, S. Caporusso, and M. Siegel, "Cyber range for industrial control systems (cr-ics) for simulating attack scenarios." in *ITASEC*, 2021, pp. 246–259.
- [11] A. Santorsola, A. Migliau, and S. Caporusso, "Reinforcement learning agents for simulating normal and malicious actions in cyber range scenarios." in *ITASEC*, 2022, pp. 1–16.
- [12] R. Beuran, "Cyber ranges," in *Cybersecurity Education and Training*. Springer, 2025, pp. 221–259.
- [13] M. Ciccaglione, L. Bracciale, P. Loreti, and A. M. Zanon, "Cyber range for space systems: Training scenarios for satellite cybersecurity preparedness," 2025.
- [14] European Space Agency. (2025) Estonia to host europe's new space cybersecurity testing ground. European Space Agency. Accessed: 2025-06-20. [Online]. Available: https://www.esa.int/Applications/Connectivity_and_Secure_Communications/Estonia_to_host_Europe_s_new_space_cybersecurity_testing_ground
- [15] J. Branham, J. Englander, J. Evans, B. Hughes, M. McIntyre, W. Sanders, J. Springmann, and C. Work, "NOS3: NASA Operational Simulator for Small Satellites," NASA, Tech. Rep. NASA/TP-2019-220266, 2019, accessed: 2025-06-20. [Online]. Available: <https://ntrs.nasa.gov/citations/20190001386>
- [16] A. Costin, H. Turtiainen, S. Khandker, and T. Hämäläinen, "Towards a unified cybersecurity testing lab for satellite, aerospace, avionics, maritime, drone (saamd) technologies and communications," *arXiv preprint arXiv:2302.08359*, 2023.
- [17] F. Patrone, P. Loreti, L. Fiscariello, L. Bracciale, A. Amici, A. Detti, C. Roseti, F. Zampognaro, M. Luglio, G. Bianchi *et al.*, "Opensatrange: An open cyber range for operators and users of satellite communication networks," in *CEUR WORKSHOP PROCEEDINGS*, vol. 3731. CEUR-WS, 2024.
- [18] VMware, Inc., *VMware ESXi: Bare Metal Hypervisor*, 2023. [Online]. Available: <https://www.vmware.com/products/esxi-and-esx.html>
- [19] Microsoft Corporation, *Hyper-V Overview*, 2023. [Online]. Available: <https://learn.microsoft.com/en-us/virtualization/hyper-v-on-windows/>
- [20] Red Hat, Inc., *Kernel-based Virtual Machine (KVM)*, 2023. [Online]. Available: <https://www.linux-kvm.org/>
- [21] Proxmox Server Solutions GmbH, *Proxmox Virtual Environment*, 2024. [Online]. Available: <https://www.proxmox.com/en/proxmox-ve>
- [22] "Tm synchronization and channel coding. recommendation for space data system standards," CCSDS, Washington, D.C., Blue Book CCSDS 131.0-B-3, September 2017.
- [23] "Radio frequency and modulation systems—part 1: Earth stations and spacecraft," CCSDS, Washington, D.C., Blue Book CCSDS 401.0-B-30, February 2020.
- [24] "Tm synchronization and channel coding - summary of concept and rationale," CCSDS, Washington, D.C., Green Book CCSDS 130.1-G-3, June 2020.
- [25] "Tc synchronization and channel coding. summary of concept and rationale," Consultative Committee for Space Data Systems (CCSDS), Washington, D.C., Green Book CCSDS 230.1-G-2, November 2012, issue 2.
- [26] "Tc synchronization and channel coding. recommendation for space data system standards," Consultative Committee for Space Data Systems (CCSDS), Washington, D.C., Blue Book CCSDS 231.0-B-3, September 2017, issue 3.
- [27] MathWorks, "End-to-end ccscds telecommand simulation with rf impairments and corrections," 2023, <https://it.mathworks.com/help/satcom/ug/end-to-end-ccscds-telecommand-simulation-with-rf-impairments-and-corrections.html>, Accessed: 2025-06-24.
- [28] MathWorks, "End-to-end ccscds telemetry synchronization and channel coding simulation with rf impairments and corrections," 2023, <https://it.mathworks.com/help/satcom/ug/end-to-end-ccscds-telemetry-synchronization-and-channel-coding-simulation-with-rf-impairments-and-corrections.html>, Accessed: 2025-06-24.
- [29] "Propagation data and prediction methods required for the design of earth-space telecommunication systems," International Telecommunication Union, Radiocommunication Sector (ITU-R), Recommendation ITU-R P.618-13, December 2017, https://www.itu.int/dms_pubrec/itu-r/rec/p/R-REC-P.618-12-201507-S!!PDF-E.pdf Version 13, Geneva.
- [30] "Meteostat: Historical weather and climate data," <https://meteostat.net>, 2024, accessed: 2025-06-24.
- [31] MathWorks, "p618propagationlosses: Itu-r p.618 propagation loss model," <https://it.mathworks.com/help/satcom/ref/p618propagationlosses.html>, 2023, accessed: 2025-06-24.
- [32] "Recommendation itu-r s.1525—impact of interference from the sun on earth-satellite links," ITU Radiocommunication Sector (ITU-R), Geneva, Recommendation S.1525, July 2001, https://www.itu.int/dms_pubrec/itu-r/rec/s/R-REC-S.1525-1-200209-I!!PDF-E.pdf.
- [33] NASA, "Concept of operations for the space link extension services," National Aeronautics and Space Administration, Tech. Rep. NASA/TM-2014-218530, 2014, accessed: 2025-06-24. [Online]. Available: <https://ntrs.nasa.gov/api/citations/20140017058/downloads/20140017058.pdf>
- [34] V. Technologies, "Sle provider implementation (modified)," <https://github.com/visionspacetec/sle-provider>, 2023, accessed: 2025-06-24.
- [35] LibreCube, "python-sle: Ccsds sle protocol stack for python," <https://gitlab.com/librecube/lib/python-sle>, 2023, accessed: 2025-06-24.
- [36] Spacepackets, "spacepackets: A python library for ccscds packet construction," <https://github.com/spacepackets/spacepackets>, 2024, accessed: 2025-06-24.
- [37] B. Fitzgerald and K.-J. Stol, "Continuous software engineering: A roadmap and agenda," *Journal of Systems and Software*, vol. 123, pp. 176–189, 2017.
- [38] N. Imagery and M. Agency, "Department of defense world geodetic system 1984: Its definition and relationships with local geodetic systems," [https://earth-info.nga.mil/php/download.php?file=coord-wgs84,2000,nIMA TR8350.2](https://earth-info.nga.mil/php/download.php?file=coord-wgs84,2000,nIMA%20TR8350.2), Third Edition, January 2000, Accessed: July 2025.