ODARIS - AN APPROACH FOR NEAR REAL-TIME BI-DIRECTIONAL COMMUNICATION IN SATELLITE OPERATIONS

D. Herschmann*, K. Schwenk*, M. Fragoso Garrido de Matos Lino*
* German Aerospace Center (DLR), German Space Operations Center (GSOC), Münchner Str. 20, 82234 Wessling, Germany

Abstract

This paper presents insights into the seamless integration of the ODARIS system into satellite operations. It highlights typical challenges and requirements from an operator's perspective for such systems. An on-board satellite experiment utilizing the ODARIS system is currently being prepared in the context of the SeRANIS mission, coordinated by the University of the Bundeswehr in Munich. For an operational proof-of-concept, the ODARIS communication system will also support a parallel experiment focused on monitoring satellite telemetry data. In summary, this paper provides an overview of ODARIS and its potential applications in satellite operations, outlines typical integration challenges and requirements, and presents initial findings from the preparation phase of an upcoming satellite experiment scheduled for the end of 2026.

Keywords

real-time communication, satellite operations, embedded systems, software development

1. INTRODUCTION

The rapid increase of satellite missions and the deployment of large-scale constellations have significantly expanded the capabilities of space-based research and commercial services, but have also intensified the demand for higher operational reliability. As mission complexity and the number of satellites increase, continuous status monitoring and minimal time-to-action have become critical for ensuring safe and efficient operations. Traditional telemetry and telecommand (TM/TC) architectures, however, face inherent limitations in this evolving landscape. Ground segment operations are typically constrained by fixed communication windows. Low- to mid-sized missions, with limited budgets, often have only a few operational passes per day or week available.

This intermittent connectivity introduces significant latency, which delays critical decision-making and reduces operational responsiveness. Consequently, the need for cost-effective real-time communication links between spacecraft and ground operators has emerged as a fundamental requirement for modern space missions.

To address these challenges, this paper presents a comprehensive system for real-time satellite communication tailored to resource-constrained missions. The subsequent sections first provide an overview of typical operator requirements, emphasizing the need for low-latency data delivery and streamlined command execution. Subsequently, the technical architecture of the "On-board Data Analysis And Real-Time Information System" (ODARIS) platform is introduced, along with a description of the operational in-orbit experiment OPC-4. It leverages the platform's capabilities for real-time TM monitoring and enables rapid operator feedback. The paper concludes with a summary of key findings and outlines a pathway for future platform evolution.

2. REQUIREMENTS FOR SPACECRAFT OPERATIONS

The operations of Low Earth Orbit satellites are typically constrained by the availability of uplink and downlink streams. The number of daily contacts depends on mission-specific requirements, but inevitably results in extended periods of no visibility between ground stations and spacecraft.

This introduces significant latency between on-board events and their subsequent acknowledgment from the ground segment. Moreover, any corrective response to such events is similarly delayed. This includes the resolution of on-board anomalies, as well as the correction of uploaded timelines or payload operation parameters. Prolonged response times can increase the number of occuring on-board anomalies and extend mission outages, resulting in financial and operational penalties on the ground segment. Furthermore, expanding the ground station network or purchasing additional contact passes incurs significant additional operational costs for the mission.

Equipping satellites with communication terminals that enable real-time connectivity to the ground segment via a relay system, such as a satellite con-

CC BY 4.0

doi: 10.25967/680460

stellation with extended coverage capable of reliably forwarding bidirectional data, offers a cost-effective solution. This approach enables an integrated space-ground architecture where ground-based automated systems can leverage high-performance computing and auxiliary data to rapidly respond to on-board events. For instance, an on-board anomaly can be acknowledged immediately, allowing a recovery procedure to be uplinked under strictly defined operational boundaries, ensuring optimal resolution. For novel or unanticipated events lacking predefined responses, 24/7 ground personnel can rapidly develop and deploy new procedures.
This capability enables near real-time synchronization between the space and ground segments in response to ground status changes or new payload requests. For example, urgent imaging requests triggered by terrestrial events can be uplinked without delay. Additionally, ground-side operational changes, such as a ground station outage, can be seamlessly integrated into the on-board execution timeline prior to activation. The OPC-4 experiment, hosted on the ODARIS platform and leveraging the system's real-time relay capabilities, is designed to demonstrate the operational benefits of this architecture for ground segment operations.

3. THE ODARIS PLATFORM

The foundation for the advanced communication interface is the ODARIS platform, designed for integration into "Low Earth Orbit" (LEO) satellite missions. The system is currently under development at DLR. This chapter provides an overview of the platform's core concept, focusing on its real-time communication hardware and software components. For further details on other aspects of the platform, please refer to earlier publications on ODARIS [1, 2].

3.1. Experiment Framework Overview

The design enables seamless integration as a payload unit into any satellite platform. From a software perspective, it is developed as application software compatible with most Linux-based operating systems. The system has been ported to various modern embedded hardware architectures, including ARMv7, ARMv8, and RISC-V. As a continuous development and testing environment across these architectures, popular "Commercial Off-The-Shelf" (COTS) devices such as the NVIDIA Jetson [3] and PolarFire SoC [4] are utilized. ODARIS will be validated as a research experiment aboard the upcoming SeRANIS mission [5], coordinated by the University of the Bundeswehr. The experiment is scheduled for commissioning on the satellite by the end of 2026.

3.2. Services Concept

The platform comprises an autonomous data processing and monitoring component, as well as a real-time information system. The autonomous data processing focuses on AI-based ship detection as a proof-of-concept scenario. This functionality was integrated as the "Yolo-ODARIS Ship-detection Interface" (YOShI) module and was presented in a previous publication [6].
This paper focuses on the information system. At its core, the system follows a database-driven architecture where all platform data is stored in structured tables. SQLite [7] was selected as the central data management component due to its proven long-term reliability on edge devices. The software architecture employs a modular service-oriented design, which is illustrated in Figure 1.

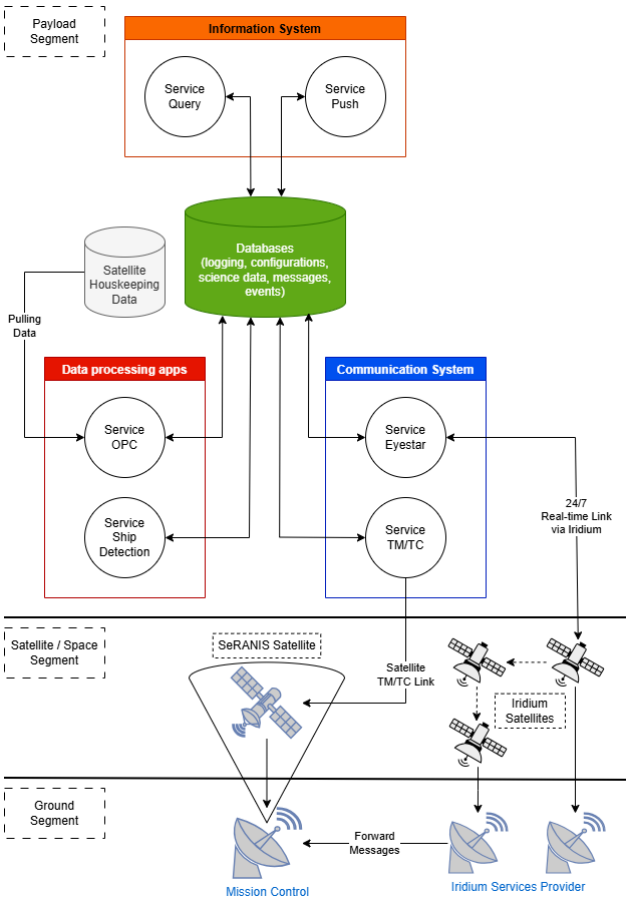


FIG 1. ODARIS ecosystem in SeRANIS Mission

Each service operates within its own execution cycle and domain, with data and information exchange occurring solely through database tables. This enables a fully decoupled, flexible, and asynchronous communication framework that forms the basis of the real-time communication system between ODARIS and ground operators.

3.3. Communication System

The ODARIS real-time communication system establishes a bi-directional link that complements the standard TM/TC link provided by the satellite platform. Such a space link can be established in two ways:

- Direct transmission from the satellite to multiple ground stations, which are distributed globally.

CC BY 4.0

- Relay transmission via a satellite constellation to a single or limited number of ground stations operated by a service provider.

Each architecture offers distinct advantages and disadvantages, depending on the specific use cases and mission scope. A key objective of ODARIS is to enable near real-time communication for small- to medium-sized satellite projects with limited budgets. Such budget constraints typically preclude the leasing of multiple ground stations to ensure continuous coverage across all orbital positions. In contrast, utilizing a satellite constellation service provider is more cost-effective. This approach shifts costs and resources from managing multiple ground station contracts to a simplified, data-driven model, where infrastructure expenses are bundled into a per-byte pricing structure.

Based on prior studies, three satellite constellations have proven suitable [8, 9]:

- Globalstar [10]
- Orbcomm [11]
- Iridium [12]

Rapid advancements in large-scale satellite constellations may expand this list, including Starlink [13] and IRIS^2 [14].

The primary focus of ODARIS lies in the platform's software architecture and high portability. For this experiment, a commercial off-the-shelf (COTS) device was selected to enable bi-directional end-to-end communication in orbit. The American company NearSpaceLaunch [15] provides the Eyestar-S4 radio device [16], which facilitates communication via the Iridium network. Its reliable in-orbit functionality has been demonstrated in previous studies [17, 18]. The device operates in the L-band spectrum and utilizes the Short Burst Data (SBD) functionality of the Iridium network [19]. A dedicated ODARIS service handles incoming and outgoing messages between the Eyestar-S4 and the information system. Due to the overall services structure, messages have a hard limit of 200 bytes, which directly influences the design of the ODARIS messaging protocol in the information system.

3.4. Information System

Derived from the database-centric and service-oriented architecture, the core concept of the information system centers on its messaging framework. All services communicate by reading and writing messages to database tables and addressing other services via the messaging protocol. This architecture resembles an email system with dedicated inboxes and outboxes for each service. The information system provides two main functionalities for operators and end users, e.g. scientists, and can be accessed from anywhere and anytime on Earth:

- Querying any type of data stored on the platform.
- Configuring scheduled and/or trigger-based events to reliably gather information in form of short push notifications.

Dedicated services are assigned to each of these functionalities.

The Query service processes user requests, which may be formulated as raw SQL statements. Upon receiving a request in its designated inbox, the service executes the query and stores the result in a response message directed to the ground segment. This service is also responsible for defining scheduled or trigger-based push events.

The Push service processes all registered events. The event list is stored in a structured table and can be created, modified, or deleted by operators at any time during the mission. Event definitions follow a fixed schema, as detailed in Table 1:

TAB 1. Push Event Table Schema

Column	Type	Description
`PushQueryId`	int	Unique event identifier
`Start`	text	Activity window start time
`Stop`	text	Activity window stop time
`Period_s`	int	Check period in seconds
`Priority`	int	Message priority
`RetChan`	int	Message return channel
`Category`	int	Message category
`Db`	text	Path to the SQLite database file
`AnswerCnt`	int	Event execution counter
`Query`	text	SQL query to execute
`LastExecuted`	text	Timestamp of last execution

Descriptions of schema elements:

- **PushQueryId**: Unique identifier to assign received messages to the correct event definition
- **Start**: Timestamp when the event is active.
- **Stop**: Timestamp after the event is inactive.
- **Period_s**: Period when the event is triggered for execution.
- **Priority**: Defines the execution order when multiple events are active simultaneously.
- **RetChan**: Specifies the communication channel to which response messages are returned (e.g., satellite TM/TC or real-time communication system).
- **Category**: Definition is optional but useful when managing many similar events.
- **Db**: Enables execution of event tasks outside the standard ODARIS database, e.g., when a third-party application stores data in a separate file.
- **AnswerCnt**: Tracks the number of times the event has been executed.
- **Query**: Contains the actual event request or task, and may include conditions to enable trigger-based notifications.
- **LastExecuted**: Can be monitored by operators for operational tracking.

Upon completion, both services return an appropriate response message to the operator. The messaging protocol itself is detailed in the following section.

3.5. Messaging Protocol

Over time, several communication standards have been developed and refined through operational experience in space missions. A widely adopted

CC BY 4.0

standard is ECSS-E-ST-70-41C, which defines the CCSDS packet format [20]. The protocol structure comprises multiple header fields, as summarized in Table 2:

TAB 2. CCSDS PUS-C Packet Structure Overview

Component	Size	Description
`Primary Header`	6 B	Version, type, APID, ...
`Sec.TC Header`	5 B	Service ID, subservice ID, ...
`Sec.TM Header`	13 B	Common fields + timestamp
`Data Field`	var.	Service-specific payload
`Checksum`	2 B	Packet error control

The complexity of typical space missions necessitates multi-component header structures. However, due to the strict payload limitations of the Iridium network, these standard protocols are unsuitable for the ODARIS messaging system. Consequently, a custom lightweight protocol was developed to minimize header overhead. Typical for communication protocols, the ODARIS messaging format consists of a fixed header and a variable-length data field, as shown in Table 3:

TAB 3. ODARIS Messaging Protocol Format

Component	Size	Description
`MsgId`	1 B	Message identifier
`Category`	1 B	Message category
`Channel`	1 B	Addressed service ID
`Priority`	1 B	Message priority level
`RetChan`	1 B	Return channel selection
`Data`	var.	Payload/content

Descriptions of protocol elements:

- **MsgId**: Assigns a unique identifier to each message.
- **Category**: Definition is optional but useful when managing many similar queries.
- **Channel**: Specifies the target service within ODARIS.
- **Priority**: Defines the execution order when multiple events are active simultaneously.
- **RetChan**: Specifies the communication channel to which response messages are returned (e.g., satellite TM/TC or real-time communication system).
- **Data**: Contains the actual user query or response payload.

Queries and Push Event notifications share the same Message ID with their corresponding responses, enabling correct correlation upon reception. The protocol payload is initially formatted as a text string and subsequently serialized for satellite-to-ground transmission. If a response exceeds the maximum payload size, ODARIS automatically fragments the data into multiple messages using a multi-message transmission scheme. A sequence counter is prepended to the header to maintain message ordering.

4. THE OPC-4 EXPERIMENT

The OPC-4 experiment is coordinated by the "German Space Operations Center" (GSOC) at DLR. The primary objective is to implement and validate an innovative in-flight mission operations concept, designed to enhance operational autonomy, while strictly adhering to established satellite operation standards and mission control requirements. The integration of the ODARIS real-time communication system is intended to provide the operations team with near-instantaneous access to critical housekeeping data, e.g. Attitude and Orbit Control System (AOCS) telemetry. elemetry data shall be accessible via both manual requests and automated push notifications triggered by onboard anomalies or critical events. To ensure continuous monitoring of the satellite system, periodic live status updates shall be transmitted. Furthermore, the experiment will evaluate whether the operations team can transmit authenticated telecommands to the satellite via the ODARIS messaging framework from within the mission control center.

The experiment is implemented as a dedicated service within the standard ODARIS architecture, as illustrated in Figure 1. Communication through the messaging system is facilitated via shared SQLite database tables. This architecture enables the bi-directional storage and forwarding of both telecommands and telemetry data, either to the satellite platform or to ground operators in near real-time.

Test telemetry data provided by the satellite platform to the experiment payload unit is stored in MySQL databases [21] and periodically accessed by the OPC service. Through an internal configuration framework, specific telemetry parameters can be queried at predefined intervals and automatically forwarded to the ground segment via the ODARIS real-time communication link.

5. CONCLUSION AND WAY FORWARD

This paper presented a near real-time communication concept leveraging global satellite networks to address the growing operational challenges posed by increasing satellite mission counts and constellation deployments. The operator requirements outlined in Section 2 highlight that low-latency data delivery and streamlined command execution are critical for maintaining mission reliability across large-scale satellite operations.

The ODARIS platform fulfills these requirements by enabling a continuous monitoring architecture that overcomes the limitations of traditional TM/TC ground segment operations constrained by fixed communication windows. A key contribution is the custom messaging protocol, which achieves significant message size optimization through compressed bit-level formatting compared to conventional standards such as PUS. This efficiency enables practical real-time telemetry delivery within the narrow bandwidth available on typical small satellite links.

CC BY 4.0

The OPC-4 experiment is designed to validate this approach by demonstrating real-time TM monitoring using the platform's capabilities during the upcoming SeRANIS mission, scheduled for the end of 2026. It will demonstrate the end-to-end communication chain and provide operational data to assess the concept's feasibility for routine mission support.
Building on this foundation, several enhancements are planned for ODARIS to increase the platform's maturity. The messaging protocol will be further optimized through refined compressed bit-level encoding to maximize throughput efficiency. An encryption module will also be integrated to enable secure message transmission between the satellite platform and end-users. Additionally, a comprehensive survey of commercial satellite constellations such as Starlink and IRIS^2 will evaluate the availability of suitable COTS communication hardware and assess the effort required for customized transceiver designs.

Contact address:

daniel.herschmann@dlr.de

References

[1] Kurt Schwenk and Daniel Herschmann. Onboard data analysis and realtime information system - status & outlook. In *Deutscher Luft- und Raumfahrtkongress 2022 (DLRK 2022)*, September 2022.

[2] Kurt Schwenk and Daniel Herschmann. Onboard data analysis and real-time information system. In *Deutscher Luft- und Raumfahrtkongress 2020*, Oktober 2020.

[3] NVIDIA Corporation. Nvidia jetson modules website, 2026. https://developer.nvidia.com/embedded/jetson-modules.

[4] Microchip Technology Inc. Polarfire soc discovery kit website, 2026. https://www.microchip.com/en-us/development-tool/MPFS-DISCO-KIT.

[5] SeRANIS Redaktions-Team. Seranis public website, 2026. https://seranis.de/.

[6] Mattia Scaglioni, Daniel Herschmann, and Kurt Schwenk. Yoshi: Yolo-odaris ship-detection interface - an innovative approach to on-board data analysis for ship-detection. In *European Data Handling & Data Processing Conference 2025 (EDHPC 2025)*, Oktober 2025.

[7] SQLite Project. Sqlite project homepage, 2026. https://sqlite.org/.

[8] Andrew D. Santangelo and Paul Skentzos. Utilizing the globalstar network for satellite communications in low earth orbit. In *54th AIAA Aerospace Sciences Meeting, San Diego, California, USA*, 2016. DOI: 10.2514/6.2016-0966.

[9] Stefan Schönegger. Investigation of leo communication networks for space telemetry and commanding. *diploma thesis - Fachhochschule Technikum Wien and DLR*, 2004.

[10] GlobalStar Inc. Globalstar website, 2026. https://www.globalstar.com.

[11] Orbcomm Inc. Orbcomm website, 2026. https://www.orbcomm.com.

[12] Iridium Communication Inc. Iridium, 2026. https://www.iridium.com.

[13] SpaceX. Starlink website, 2026. https://www.starlink.com/technology.

[14] EUSPA. Iris² website, 2026. https://www.euspa.europa.eu/eu-space-programme/secure-satcom/iris2.

CC BY 4.0

[15] NearSpace Launch. Nearspace launch company website, 2026. https://www.nearspacelaunch.com.

[16] Nearspace Lauch. Nearspacelaunch eyestar product, 2026. https://nearspacelaunch.com/eye-star/.

[17] Jeff Dailey, Matthew Orvis, John Pugsley, Matthew Voss, and Hank Voss. Launch & early operations phase (leop) iridium eyestar-s4 24-7 link with mosaic-x5 gps and launch results. In *38th Annual Small Satellite Conference*, 2024.

[18] Jeff Dailey, Matthew Orvis, John Pugsley, Matthew Voss, Mike Schuckel, Nathan Turner, and Hank Voss. Pushing the boundaries with the thinsat and troop platforms: F2 flight results and testing new systems. In *39th Annual Small Satellite Conference*, 2025.

[19] Iridium Communications Inc. Iridium short burst data (sbd), 2026. https://www.iridium.com/services/iridium-short-burst-data-sbd.

[20] European Cooperation for Space Standardization. Space engineering - spacecraft command and telemetry (c&t) packet structure. Technical Report ECSS-E-ST-70-41C, ECSS, 2026.

[21] MySQL Project. Mysql project homepage, 2026. https://www.mysql.com/.

CC BY 4.0