17th International Conference on Space Operations, Dubai, United Arab Emirates, 6 - 10 March 2023.

SpaceOps-2023, ID # 353

Analysis and Impact of the End-to-End Communication Chain on a DLR1 Real Time On-Orbit Servicing Mission Project

Dominic Lucas[a], Rossella Falcone[b], Christian Stangl[c], Rainer Krenn[d], Bernhard Brunner[e], Felix Huber[f]

[a] *Ground Data Systems Engineer, Communication and Ground Stations, DLR*, dominic.lucas@dlr.de
[b] *Ground Data Systems Manager, Communication and Ground Stations, DLR*, rossella.falcone@dlr.de
[c] *TM/TC System Engineer, Mission Technology, DLR*, christian.stangl@dlr.de
[d] *Head of Space Systems Group, Institute of System Dynamics and Control, DLR*, rainer.krenn@dlr.de
[e] *Robotics Software Architect, Institute of Robotics and Mechatronics, DLR*, bernhard.brunner@dlr.de
[f] *Institute Director, Space Operations and Astronaut Training, DLR*, felix.huber@dlr.de

Abstract

The German Aerospace Center, DLR, is researching and developing technologies to enable future robotic On-orbit Servicing (OOS) missions. Current studies are being undertaken by the RICADOS (Rendezvous, Inspection, Capturing and Detumbling by Orbital Servicing) project. RICADOS traces its heritage to the proposed Deutsche Orbitale Servicing (DEOS) mission. The project employs a holistic approach to developing the key technologies and operational procedures for OOS missions in a collaborative effort between multiple DLR Institutes. It encompasses several disciplines: camera and visual technologies for inspection, guidance, navigation and control (GNC) purposes, and real-time telerobotic operations. Furthermore, a key aspect of the project is the integration of operations into the operational German Space Operation Center (GSOC) mission control infrastructure. This enables conducting end-to-end tests from the satellite and robotic control consoles, across the simulated ground and space communication chain, to the (simulated) satellite bus, employing the tools and procedures that are used by real missions. Additionally, hardware-in-the-loop operations are conducted using the European Proximity Operations Simulator (EPOS 2.0) and On-Orbit Servicing Simulator (OOS-Sim) facilities at the DLR Space Operations and Astronaut Training Institute and Robotic- and Mechatronics Institute, respectively, in Oberpfaffenhofen (Munich, Germany). This allows real-time mission operations focusing on testing and verification of complex rendezvous, inspection, and capturing manoeuvres.

Carrying out real-time telerobotic activities with haptic-feedback during these operations impose constraints to the communication chain and protocols regarding packet loss, round-trip delay, and jitter. One approach is to deal with these restrictions by implementing dedicated communication links for conducting robotic operations, which would not allow simultaneous TM/TC operations. RICADOS manages to conduct both standard spacecraft TM/TC and robotic operations by multiplexing the independent robotic payload and spacecraft bus TM/TC data streams into a single physical communication link. This offers the ability to independently and simultaneously process the robotic payload and spacecraft bus control data. The multiplexing is realized using a FPGA (Field Programmable Gate Array) device. Furthermore, it enables maintaining the strict timing requirements for real-time operations as the FPGA uses a fixed clock. The ground segment and communications link are simulated by the Communication System (COSY). COSY handles conversion and homologation of the space protocols, multiplexing of spacecraft and robotic data streams, and simulation of the communication link via a wide-area-network ground and space link.

This paper reports on the testing and evaluation of the COSY communication chain, following four years of successful implementation and operation. The testing and analysis aimed to characterize the delay and jitter across the COSY system and its sub-systems, with a view to determining compliance with the RICADOS project's latency and jitter requirements of 100 ms and 2.5 ms, respectively. The results of the testing and analysis indicated that cross-device latency and jitter meet the requirements in relevant cases, setting the stage for further optimization and development of the COSY system.

Keywords: OOS, Teleoperation, Communication, Ground Segment, GSOC, DLR

1. Introduction

1.1 Communications for OOS

Space debris poses a threat to the use of earth orbits for satellite missions and requires strategies for removal or deorbiting. One solution is the use of telerobotic missions, where a human operator controls a robot in real-time, for satellite servicing or deorbiting debris. However, there is currently no standardized interface for performing telerobotic operations with any ground station. The ROKVISS [1] and Kontur-2 [2] missions have demonstrated the feasibility of

telerobotic operations, but their solutions are specialized and not suitable for general use. A universal interface that can be used with any ground station and accommodate various communication technologies is needed for regular telerobotic missions and station handovers. To support these missions, mission control centres, mission users, and service providers need to consider the capabilities of their ground stations and determine if existing procedures and technologies are sufficient. There is a trade-off between investing in more expensive technical solutions and implementing simpler, cost-effective operational workarounds that rely on human dependability. Communication is critical for robotic operations and requires real-time or near real-time connectivity. Existing telecommunication service providers are transitioning to IP switching networks, which may not be suitable for real-time space communication. The impact of delays, jitter, and packet loss on the performance of the system must be considered to ensure reliable communication for robotic operations. [3]

1.2 RICADOS

RICADOS (Rendezvous, Inspection, Capturing, and Detumbling by Orbital Servicing) is a joint project between Space Flight Operations and Astronaut Training (RB), the Institute for Robotics and Mechatronics (RM), the Institute for System Dynamics and Control (SR), and the Institute for Optical Sensor Systems (OS) with the aim of developing on-board and ground-segment systems, technologies, and operational procedures necessary to facilitate autonomous inspection, approach, rendezvous, and robotic and telerobotic capabilities for future on-orbit servicing (OOS) missions. The RICADOS project traces its origins to the DEOS (DEutsche Orbitale Servicing) proposed in 2012 [4] and continues the work of the technologies developed for the mission.

The RICADOS system is a unique set of hardware and facilities that are used to perform end-to-end tests for space missions. The system includes the EPOS 2.0, which simulates the space segment and inspection and rendezvous operations, and the OOS-Sim, which simulates capturing and detumbling operations. The operations are carried out in a real multi-mission control room.

The system's inspection phase starts at around 20 meters, where the system flies around the target and stereo cameras capture images. The data is then saved on-board and sent to the ground for processing, and a 6D pose estimation is performed to investigate the geometric properties of the target. A 3D point cloud model is generated and compared with a reference model of the target to assess any damage. The development and validation of the system are performed with multi-camera-IPS. Software mock-ups can also be used to simulate a diverse range of targets.

The RICADOS system's rendezvous operations use 2D/3D optical sensors such as CCD mono and 3D PMD cameras, and an edge tracking algorithm is used to determine the attitude and movement of the target relative to the servicer. The operator monitors activities and commands the satellite at a high-level during short contact phases, and the GNC system autonomously determines the approach trajectory, making it adaptable and flexible. Future systems will implement LiDAR.

Capture operations are performed using a robotic manipulator arm on the servicer. A pre-defined grasping point is selected based on a model from inspection passes, and the robot's end effector has stereo cameras to determine the relative position of the arm with respect to the target. [5]

2. Overview of the Communication Chain

The goal of the communication system is to establish a single simulated space link for delivering satellite commands in real-time, while also allowing for standard satellite operations with guaranteed delivery to occur simultaneously. To accomplish this, commands from the satellite console (SACO) are combined with commands from the robotic console (ROCO) using an FPGA-based packet merger called the Deterministic Network Gateway (DNG) or MEGI.

SASI is the central satellite simulator within the overall RICADOS mission simulator. It simulates the physics of the orbital scenario in multiple domains and provides representations of the servicer’s onboard subsystems like AOCS, power, thermal and communication. As the central node it maintains interfaces to all physically existing onboard subsystems like robotics, rendezvous and inspection, to the ground station and to the robotic motion simulators EPOS and OOS-Sim. In its role as central communication relay it is a major contributor to TM/TC roundtrip delays.

The satellite console (SACO) is a Linux-based workstation used for multi-mission satellite operations. It communicates with spacecraft using the Spacecraft Operating System (SCOS) [6], which runs on a virtual machine on a VMware ESXi Server. SCOS uses TCP/IP connections to exchange telecommands and telemetry with a ground station. In order to meet the real-time requirements for telerobotic operations, TCP/IP packets are converted to UDP/IP by a the SCOS to DNG Gateway (SDG). Since there is no actual ground station in the simulation setup, the gateway must also simulate acknowledgements for SCOS to provide information on telecommand transmission by the ground station. The robotic console (ROCO) is composed of a telepresence master device, which is based on the DLR Lightweight Robot (LWR) [7], and a software component for generating telecommands and reading telemetry. The

CCSDS Space Communication Protocols [8] [9] [10] [11] [12] are implemented using a TM/TC library written in C++11, which is the programming language used for all simulation facilities. The rendezvous console (RECO) is installed on a virtual machine on a VMware ESXi server and also uses the same TM/TC C++11 library to read telemetry and create telecommands. It interacts with the satellite's on-board guidance, navigation, and control (GNC) system. To receive telemetry and send telecommands, it shares bandwidth with SACO and uses an external interface (EXIF) of SCOS. Due to the strict packet timing requirements of the project, as well as performing standard spacecraft telecommanding and telemetry and real-time robotic operations in parallel over a single space link, it is necessary to guarantee the combination and delivery of this traffic in a deterministic way. This is realised by the implementation of an FPGA-based merger, which multiplexes and demultiplexes the respective incoming and outgoing UDP/IP packets of SDG and ROCO. The MEGI is based on a Xilinx Spartan 6 FPGA with an embedded controller for remote management. The merger takes incoming data from the uplink and downlink at a data rate of 256 kbps and 6 Mbps, and a packet rate of 200 and 400 packets/s respectively. Packets destined for uplink are multiplexed by alternately interleaving them [Figure 2.1]. SCOS and robotic packets are alternatingly sent every 2.5 ms in order to meet the 5 ms period for real-time telerobotic requirements [13]. Additionally, the merger forwards the TM data returning from SASI to SDG/SCOS and ROCO respectively. The bandwidth is limited by the reference scenario, so telecommands (CLTUs) have a maximum length of 74 octets and telemetry has a fixed length of 1115 octets. Telecommands and telemetry are forwarded at intervals of 2.5 ms.

The Communication System (COSY) facilitates both the communication between the subsystem commanding consoles, and the realistic simulation of the network, ground station, and space link. COSY implements much of the same hardware and software which is used during real missions by the DLR, and is integrated into the operational GSOC infrastructure. This allows RICADOS to perform OOS mission simulations using procedures and operating under conditions and constraints that are as close as possible to real DLR missions.

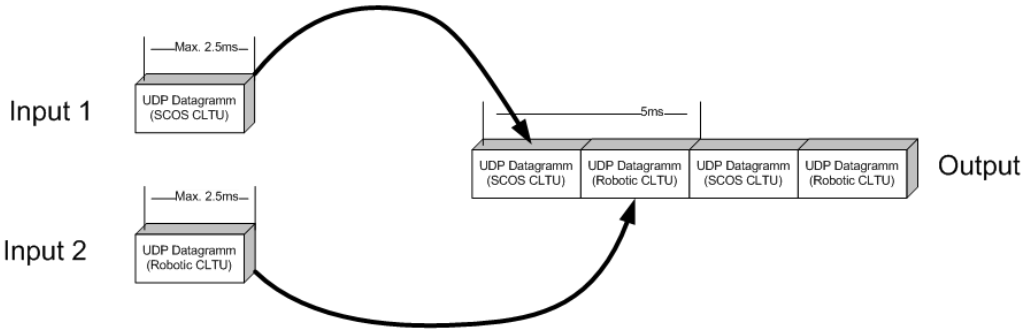

Figure 2.1 FPGA UDP interleaving [13]

They COSY consists of three main parts; the aforementioned SCOS-to-DNG-Gateway (SDG), the FPGA Merger (MEGI), and the WAN Simulator (WASI). Figure 2.2 shows an overview of the communication chain and the hardware and software configurations for SDG and WASI are given in Table 2.1 and Table 2.2 respectively. For a comprehensive description of the sub-systems and their operation, please refer to [13].

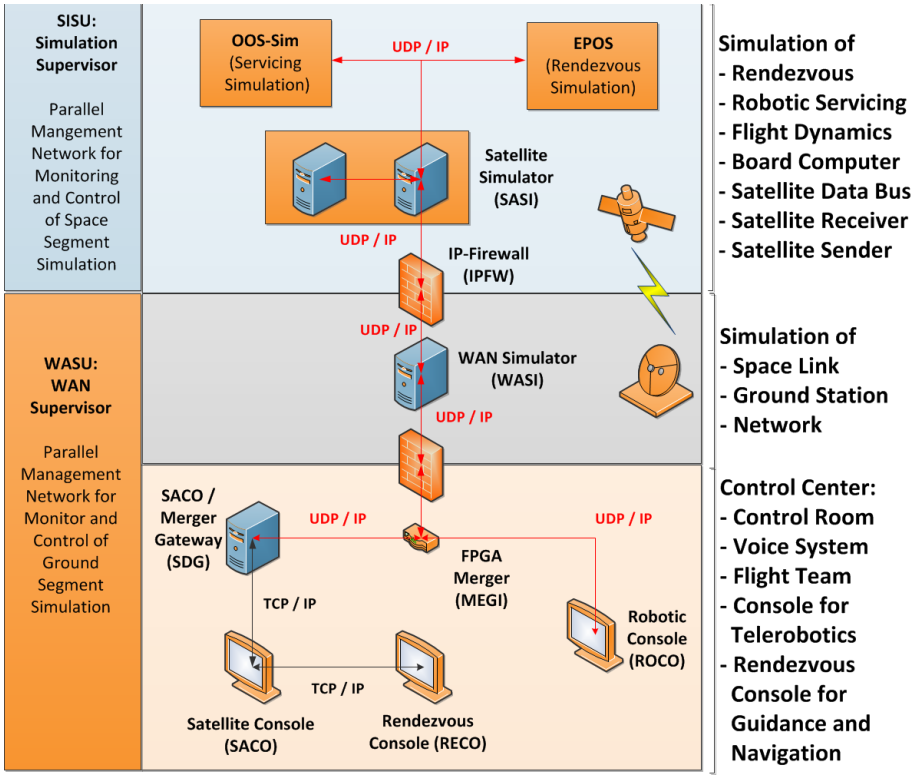

Figure 2.2 The COSY communication chain. [13]

Table 2.1 SDG hardware and software configuration.

Hardware	Comment
CPU	Intel Xeon E5-2640 v2 (2 virtual cores) @2 GHz
Memory	2,048 MB (virtual)
Network	2 x VMware VMXNET3 para-virtual network driver
Operating System	SUSE Linux Enterprise Server

Table 2.2 WASI hardware and software configuration.

Hardware	Comment
CPU	Intel Xeon X5650 6C/12T @2.67 GHz
Memory	3 x 8,192 MB Registered EEC @ 1,333 MHz
Network	Intel 82576 Gigabit Ethernet Controller (2 ports) Intel PRO/1000 PT Ethernet Controller (2 ports)
Operating System	SUSE Linux Enterprise Server

The Wan Simulator (WASI) allows the simulation of the various states of the data connection between the control room and the SC. WASI is able to set various delay and jitter values, loss, duplication and reordering of the packets. Therefore, it is possible to simulate several realistic communication scenarios (e.g. connection via GEO relay in which the round -trip delay is greater than 500 ms or an unstable internet connection with the loss of packets or duplication of the packet sequence). Figure 2.3 provides an overview of WASI GUI which can be used to manually set desired network effects. However, in practice during mission simulations, the WASI parameters are set using pre-scripted workflows which run automatically based on a timeline once triggered.

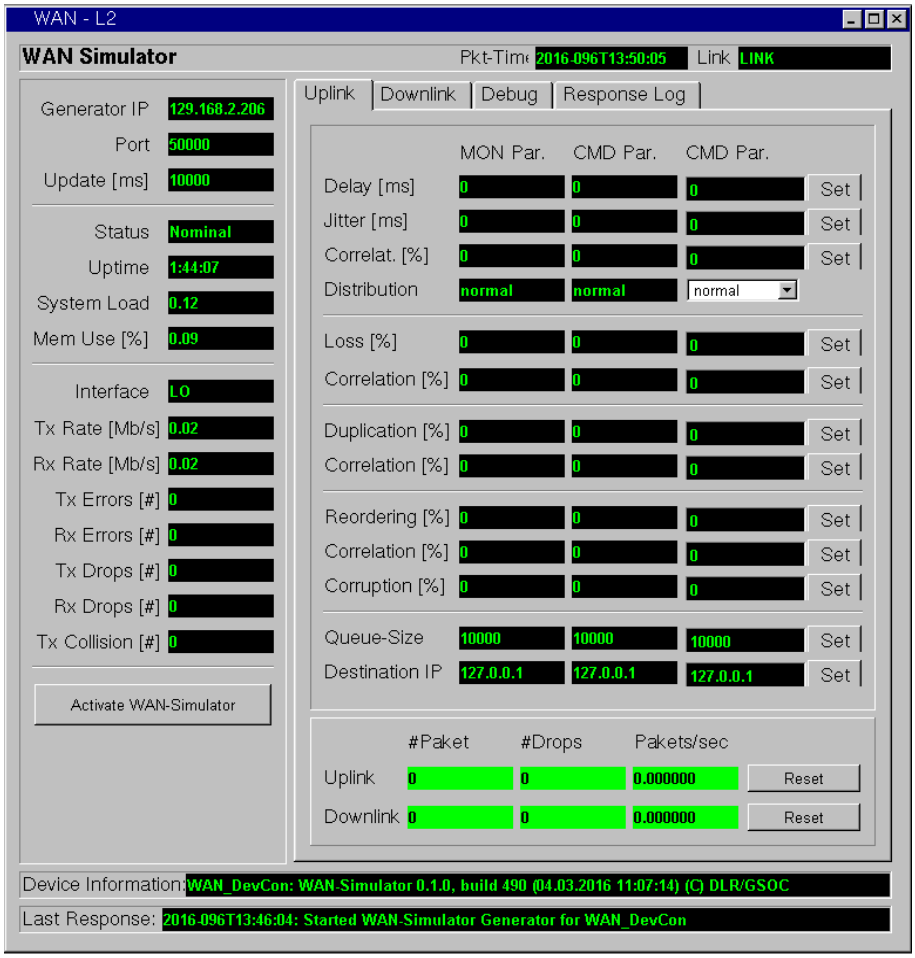

Figure 2.3 WASI control GUI. [13]

The RICADOS project places a significant emphasis on the regular execution of end-to-end mission testing, with a focus on conducting operations that closely emulate real-life mission scenarios. To accomplish this, testing and mission simulation activities are conducted within the actual operational environment and infrastructure of GSOC. Console operators are tasked with performing subsystem commanding and monitoring utilizing the same tools and conditions as those employed during actual missions. The ground-segment software employed is similar to that used for controlling the antennas at the DLR ground station in Weilheim. [14]

In order to effectively simulate an On-Orbit Servicing (OOS) mission, the tests and demonstrations are divided into separate phases and orbit passes, allowing for the simulation of a variety of mission scenarios. A typical simulated mission includes a target satellite and an OOS chaser satellite, both in a Low Earth Orbit (LEO) (Envisat), with the chaser starting at a distance of 15 meters from the target. The OOS mission encompasses two inspection passes, three rendezvous passes, and a capturing pass. The inspection and rendezvous passes are conducted during fixed direct ground station contact windows, as these are phases that necessitate reliable communications for commanding. Due to the extended duration of robotic operations, which could last far longer than typical ground station contact times of ~ 11 minutes, they are performed utilizing a Geostationary Earth Orbit (GEO) communications satellite (such as EDRS) as a relay for spacecraft and robotic communications. In between passes, a realistic loss of signal (LOS) condition is simulated by applying of a packet loss of 100% in the WASI, with additional delay and jitter values added to simulate realistic communication and resulting operating conditions for the various phases of the mission. A typical mission simulation scenario, and the corresponding WASI parameters used to influence the communications link can be seen in Table 2.3 below.

Table 2.3 Typical WASI parameters employed during a RICADOS mission simulation

Mission Phase	Duration (Minutes)	TC Packet Loss (%)	TM Packet Loss (%)	TC Delay (ms)	TM Delay (ms)
Initial LOS	3	0	100	0	0
INS 1	8	0	0	82	82
LOS	4	0	100	82	82
INS 2	8	0	0	82	82
LOS	6	0	100	82	82
RDV 1	8	0	0	82	82
LOS	4	0	100	82	82
RDV 2	8	0	0	82	82
LOS	4	0	100	82	82
RDV 3	8	0	0	82	82
LOS	4	0	100	82	82
CAP (GEO)	30	0	0	302	302

3. Testing

3.1 Aims and Objectives

The primary objective of this test campaign was to acquire an understanding of the delay and jitter characteristics of the RICADOS communication chain, specifically in regards to the terrestrial OP network. To accomplish this, both the TC and TM links were examined separately in order to establish an evaluation of the communication characteristics in each direction. The ultimate goal was to quantify the baseline delay and jitter of the end-to-end communication link, as well as to determine the effect of specific devices along the chain on these parameters. In particular, this campaign focused on the impact of SDG and WASI on these communication metrics.

3.2 Methodology

In order to accurately measure network delay between multiple systems, it is crucial to synchronize the clock on all systems in question. During the test campaign, this was achieved by synchronizing all system times with DLR Network Time Protocol (NTP) servers. Additionally, for systems where it was possible, manual synchronization with NTP servers was performed at the beginning of each testing day.

To measure the TM communication latency, several methods were used. For measuring the end-to-end latency, the secondary header in the TM CCSDS space packets was used to write the on-board system time of SASI when a space packed was created. This allows the timestamp in the secondary header to be compared to either the ERT timestamp which SDG includes when converting the UDP space packets to the TCP NCTRS packets used by SCOS to measure the simulated GS-to-SC link, or with the system time in SCOS to measure the network delay between SDG and SCOS. Additionally, SASI was also programmed to write the on-board time into the space immediately before the packet was sent over the network to simulate a “time of radiation”, which would enable measuring the time between packet generation and radiation on SASI. To record the TM packets across the communications chain, the tcpdump Linux tool [15] was used at each point with the following arguments: `tcpdump -i {interface} -s 0 -C 100 ‘src` or `dst {ip address} && port {port no.} && UDP` or `TCP’ -w file_name.pcap`. Incoming and outgoing packets were captured at SCOS, SDG, and WASI. By creating separate captures for packets entering and leaving each system, timestamps created by tcpdump at packet capture could be compared to calculate latency across each respective system.

In order to effectively compare the arrival and departure times of packets across multiple captures, its is necessary that the packets to be identifiable across the large datasets. Given the high TM rate of 400 packets per second from SASI, this task can prove to be challenging over a prolonged period of time. To mitigate this issue, the SASI TM was temporarily halted prior to the initiation of the recording process. Subsequently, the SASI TM was reactivated for a specified duration, such as 25 seconds during test 4 (Table 3.1). resulting in a more manageable number of packets for analysis. For each of this test type, each capture was checked to make sure the same number of packets had been captures. Each capture was subsequently verified to ensure that an equivalent number of packets had been captured. Under the assumption that the packets were arranged in the same order in each capture, the position of a packet in the time-sorted sequence was utilized for identification purposes. To compare packets captured at SDG and WASI, the Wireshark [16] application was utilized to open each capture file, sort the packets by their UTC capture time, and

export the packet statistics, including the capture time in the Unix epoch time format*, to CSV files. Finally, Microsoft Excel was employed to compare the Unix times of the captured packets and calculate the statistics presented in Section 5.

Table 3.1 COSY Tests

Test No.	Test Type	Comments & Observations
4	End-to-end TM	- 25 s TM tcpdump capture
6	End-to-end TM	- SCOS timestamps are SDG ERT - 18,400 packets captured
8	End-to-end TC	- Send 1,000 TC packets Packet rate limited to 20 packets/s by SDG receive and transmit acknowledgements

4. Calculations

4.1 Calculation of Cross-Device Latency

To calculate the delay across WASI and SDG, the Unix epoch timestamp generated by tcpdump when capturing an incoming packed, was subtracted from the that of the same packet exiting the respective system. The mean packet latency was calculated using the `AVERAGE` function in Excel across all the packets in each respective sample set.

4.2 Calculation of Cross-Device Jitter

The definition of packet jitter, or packet delay variability (PDV), used in this paper is that recommended by ITU-T Recommendation Y.1540, section 6.2, as the variation in packet delay with respect to a reference delay, which in the recommendation and this paper is the minimum of delay. The average jitter for each dataset was calculated by taking the mean of the delay variation of each packet with respect to the minimum packet delay.

4.3 Calculation of Median, Standard Deviation

For median values, the `MEDIAN` function of Excel was used. The standard deviation was calculated using the Excel function `STDEV.S` was used, which implements the following formula:

$$\sqrt{\frac{\sum(x-\bar{x})^2}{(n-1)}} \qquad [4.1]$$

5. Results and Discussion

5.1 Telemetry

Across WASI and SDG the mean latencies measured for TM packets in both test 4 and 6 were ~9 µs and ~750 µs respectively. The maximum and minimum latencies during test 4 were 3.9 ms and 0 ms across WASI, and 5.5 ms and ~40 µs across SDG. For test 6, the maximum and minimum values were 20 µs and 0 ms across WASI, and a ~40 µs and ~5.8 ms across SDG. The median latencies across WASI and SDG were 10 µs and 180-190 µs respectively. The latency across WASI had a standard deviation of 380 – 440 µs, and 1.07 – 1.09 ms for SDG (Table 5.1, Table 5.2, Table 5.3, Table 5.4). Furthermore, the jitter observed across both WASI and SDG, at ~9 µs and 0.7 ms respectively (Table 5.2,Table 5.4), is well below the required threshold of 2.5 ms. This low jitter value indicates that the system's performance is suitable for low latency telemetry data which is crucial for effective telerobitc operations.

In an effort to minimize the number of figures presented in this paper, the plots from the sixth test were selected as a visual representation of cross-device latencies. The data from test six was specifically chosen due to its large sample set of 18,400 packets over a 25-second time period. Nevertheless, the results obtained from test six are representative of the remaining TM test plots.

The results obtained from the WASI system presented an intriguing pattern; the calculated packet delays across WASI exhibited a discrete behaviour, with values of 0 µs, 10 µs, or 20 µs (with the exception of two outliers observed at the beginning of test 4). This step-wise variation in measured latency can be clearly observed in Figure 5.1. Furthermore, a greater proportion of packets incurred delays in the range of 0 µs to 10 µs, with a smaller proportion between 10 µs and 20 µs. One potential explanation for this phenomenon could be related to limitations in the accuracy

* The Unix epoch time is defined as the number of seconds that have elapsed since 00:00:00 UTC on 1 January 1970 [35]

of the system's timekeeping. By default, the timestamps generated by tcpdump use microsecond accuracy [17], which was the accuracy used for the test campaigns described in Section 3. However, the results from WASI suggest that the minimum resolution of time differences on WASI is 10 µs. This is noteworthy as WASI is implemented (Table 2.2) using high-resolution timers, as opposed to a virtual machine as with SDG (Table 2.1), to minimize system latencies. [18] [19]. To achieve higher precision in measuring latencies across WASI in future studies, the system would need to be configured to utilize high-resolution timers of the Linux kernel and tcpdump would need to be configured to use an appropriate high-resolution timer [20]. Additionally, it is worth considering the necessity of such precise latency tests on WASI, as its overall impact on latency and jitter is relatively insignificant compared to that of SDG (Table 2.1). However, this pursuit may have merit when considering the potential deployment of WASI as a virtual machine, utilizing latency reduction options such as the "Latency Sensitivity" options available in current virtualization platforms (such as those offered by VMware) [21].

Table 5.1 Statistics TM latencies in test 4.

Delay (ms)	WASI	SDG
Mean	0.00896	0.753
Median	0.0100	0.18
Minimum	0	0.0398
Maximum	3.94	5.52
Std. Dev.	0.0439	1.094

Table 5.2 Average jitter across WASI and SDG for test 4.

	Jitter (ms)
WASI	0.00896
SDG	0.713

Table 5.3 TM latency statistics for test 6.

Latency (ms)	WASI	SDG
Mean	0.00892	0.751
Median	0.01	0.190
Minimum	0	0.0398
Maximum	0.02	5.78
Std. Dev.	0.00382	1.074

Table 5.4 Average jitter across WASI and SDG for test 6.

	Jitter (ms)
WASI	0.00892
SDG	0.711

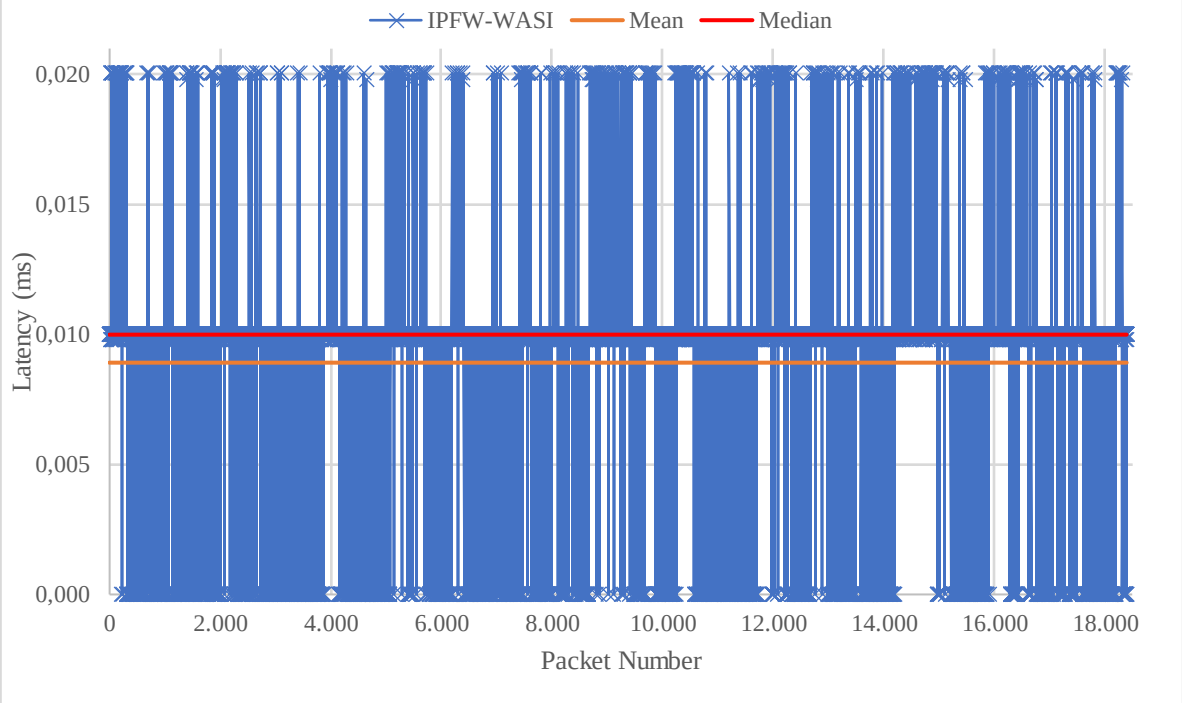

Figure 5.1 TM latency in milliseconds across WASI during test 6.

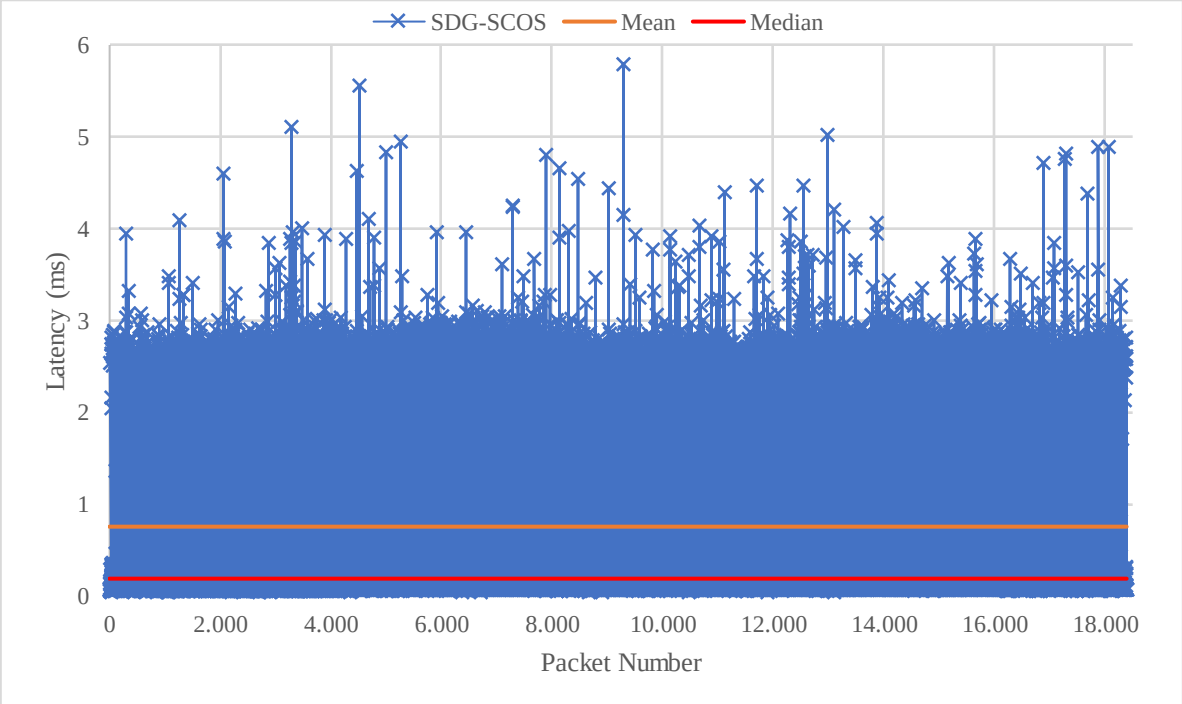

Figure 5.2 Latency in milliseconds across SDG during test 6

5.2 Telecommanding

The results of the TC test revealed notable disparities when compared to those obtained from the TM tests. The packet latency and average jitter across WASI were comparable to those observed in the TM direction. However, a significant increase in latency and jitter was observed across SDG for TC packets. As can be seen in [Table 5.5], the mean, median, minimum, and maximum latency across SDG were ~2.7 ms, ~2.7 ms, 80 µs, and ~11.6 ms, respectively. The respective end-to-end (ingress to SDG, egress from WASI) were ~9.8 ms, ~9.8 ms, 2.6 ms, 18.2 ms. It is worth noting that this high maximum latency occurred once in the 1,000-packet sample set, with the remaining packets' maximum latency values being just above 5 ms and another excursion just under 6 ms. The standard deviation of the latency across SDG and end-to-end was ~1.5 ms. The jitter across SDG was also found to be significantly higher, at ~2.59 ms and ~7 ms end-to-end [Table 5.6]. Although this is above the 2.5 ms maximum jitter required by the project, it is important to consider that SDG is only processing spacecraft commanding TCs, while the real-time robotic data bypasses SDG and is incorporated directly into the main data stream by MEGI, and as such does not pose any operational issues in that regard.

Table 5.5 TC latency statistics for test 8.

Latency (ms)	SDG	WASI	End-to-End (SDG-WASI)
Mean	2.672	0.008	9.757
Median	2.650	0.010	9.775
Minimum	0.080	0.000	2.600
Maximum	11.590	0.020	18.240
Std. Dev.	1.462	0.004	1.517

Table 5.6 Average jitter across SDG, WASI, and End-to-end for test 8.

	Jitter (ms)
SDG	2.592
WASI	0.0081
End-to-End (SDG-WASI)	7.1601

Of note is the increase in the mean end-to-end latency over time as shown by the trendline (yellow) in Figure 5.5. When restricting the latency range to that of the trendline (Figure 5.6), an increase of 0.8 ms from 9.35 ms to 10.15 ms can be seen. A possible explanation is time drift of the system clocks of SDG and WASI. CMOS clocks used in typical desktop PCs have been shown to exhibit a frequency error between 15 and 17 PPM [22]. The end-to-end PPM frequency error was calculated as 16 PPM, as shown below. This corresponds with the typical range described above, and provides valid explanation for the gradual latency increase.

$$Time\ drift = 0.8\,ms$$
$$Sample\ range = 10^3\ samples$$
$$Sample\ rate = 20\ samples/s$$
$$10^3\ samples\ \div 20\ samples/s = 50\ s$$
$$8E^{-3}\ s\ \div\ 50\ =\ 16E^{-4}$$
$$16E^{-4}\ \times\ 1E^{6}\ =\ 16\,PPM$$

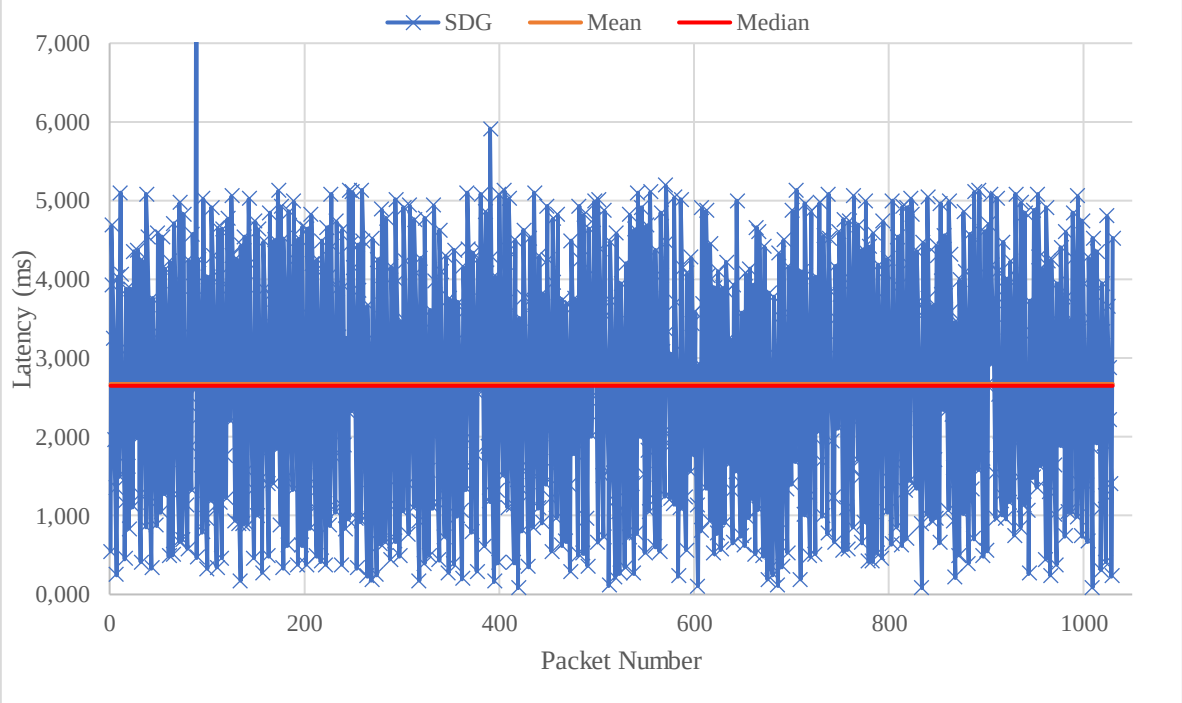

Figure 5.3 TC latency in milliseconds across SDG during test 8

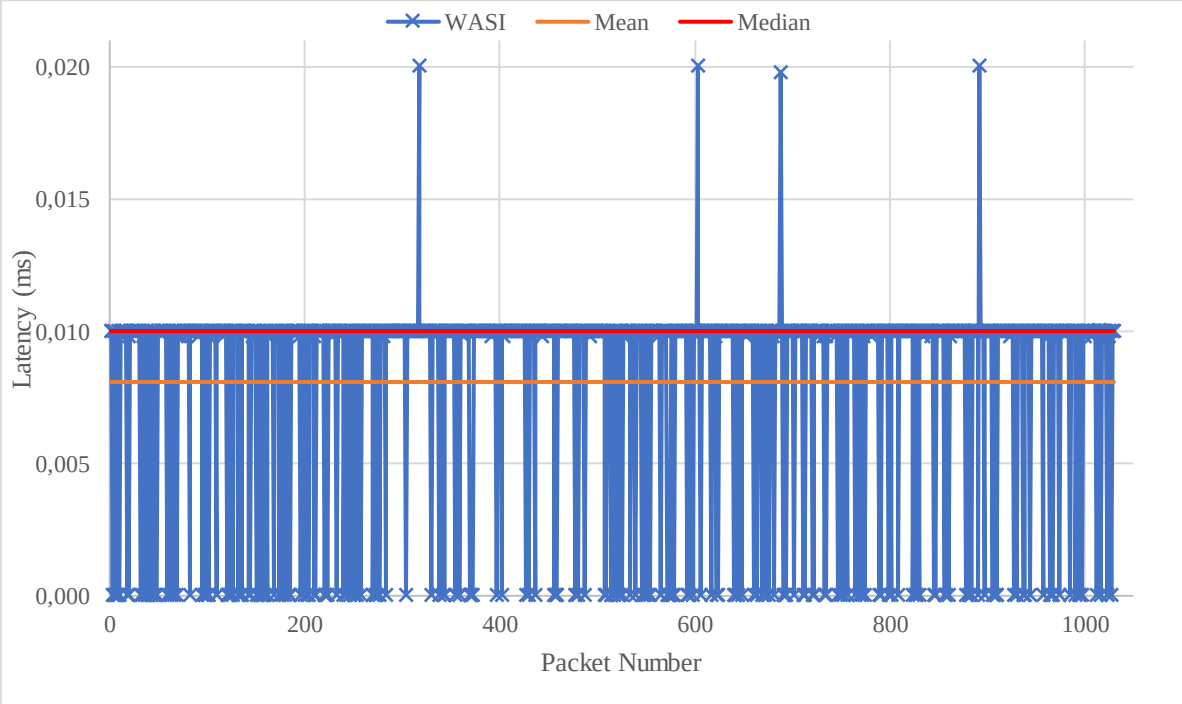

Figure 5.4 TC latency in milliseconds across WASI during test 8

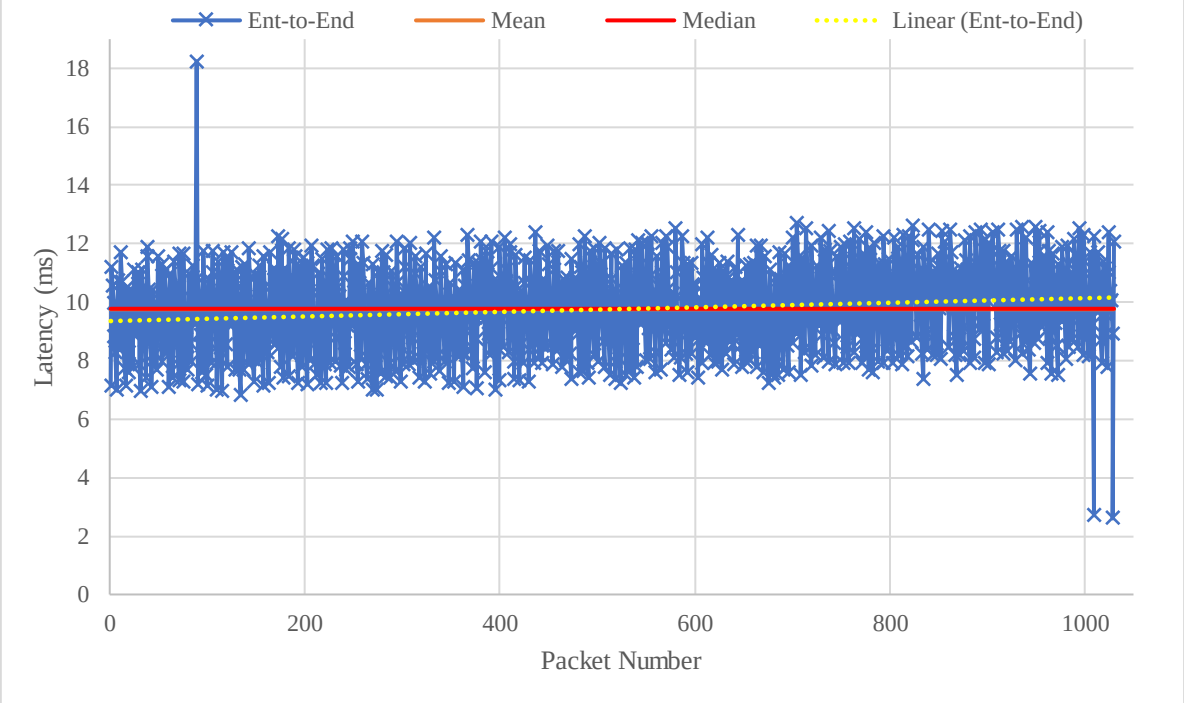

Figure 5.5 End-to-end TC latency between SDG ingress and WASI egress during test 8

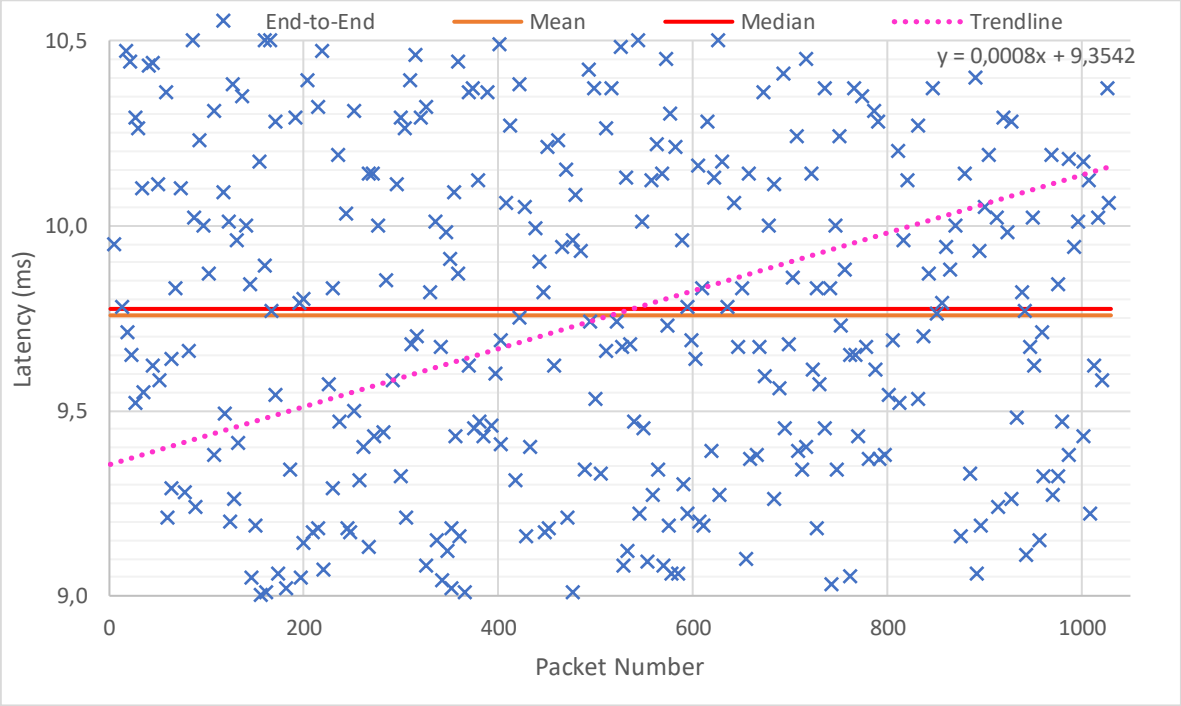

Figure 5.6 Same as Figure 5.5 but focusing on the delay range of 9 ms to 10.5 ms of the trendline.

6. Conclusions

The future of robotic space operations missions is promising, and at DLR OP, we are working to establish a reliable and functional ground station system to support them. We have focused on evaluating how our simulated ground system impacts the real-time requirements of robotic operations. Our preliminary conclusions are very encouraging. The values of jitter and latency calculated on our sub-systems in some cases are much lower than the required standards. The field of research is vast and we will continue to investigate further. Future planned work includes the

development and testing of a real-time software based merger as a redundant fullback to the hardware MEGI, as well as investigating the possibility of virtualising WASI to centralise the operational components of COSY into a single system that can flexible adapt toe the continuous development cycles of RICADOS.

Acronyms and Abbreviations

AOS	Acquisition of Signal
CCSDS	Consultative Committee for Space Data Systems
CLTU	Command Link Transmission Unit
COSY	Communication System
CPU	Central Processing Unit
CSV	Comma Separated Values
DEOS	Deutsche Orbitale Servicing
DLR	German Aerospace Center, German: Deutsches Zentrum für Luft- und Raumfahrt
DNG	Deterministic Network Gateway
EDRS	European Data Relay System
EPOS 2.0	European Proximity Operations Simulator
ERT	Earth Receive Time
EXIF	External Interface
FPGA	Field Programmable Gate Array
GEO	Geostationary Earth Orbit
GNC	Guidance, Navigation, and Control
GS	Ground Station
GSOC	German Space Operations Center
IP	Internet Protocol
IPFW	FPGA IP Firewall
ITU	International Telecommunication Union
LEO	Low Earth Orbit
LiDAR	Light Detection and Ranging
LOS	Loss of Signal
LWR	DLR Lightweight Robot
MEGI	FPGA Merger
NTP	Network Time Protocol
OOS	On-Orbit Servicing
OOS-Sim	On-Orbit Servicing Simulator
OP	DLR Site at Oberpfaffenhofen
PDV	Packet Delay Variability
PPM	Part Per Million
RICADOS	Rendezvous, Inspection, Capturing and Detumbling by Orbital Servicing
ROCO	Robotic Console
SACO	Satellite Console
SASI	Satellite Simulator
SC	Spacecraft
SCOS	Spacecraft Operating System
SDG	SCOS to DNG Gateway
TC	Telecommand
TCP	Transmission Control Protocol
TM	Telemetry
TP	Telepresence
UDP	User Datagram Protocol
UTC	Coordinated Universal Time
WASI	WAN Simulator

References

[1] G. Hirzinger, K. Landzettel, D. Reintsema, C. Preusche, A. Albu-Schäffer, B. Rebele and M. Turk, "ROKVISS - robotics component verification on ISS," *8th International Symposium on Artifical Intelligence, Robotics and Automation in Space - iSAIRAS',* 2005.

[2] M. Stelzer, B.-M. Steinmetz, B. Brunner, J. Vogel and S. Kühne, "Software architecture and design of the Kontur-2 mission," *2017 IEEE Aerospace Conference,* 2017.

[3] M. Gnat, R. Falcone, A. Hauke, A. Ohndorf and S. Eberle, "Technical and operational investigations of the real-time communication for robotic missions," *American Institute of Aeronautics and Astronautics,* 2014.

[4] G. D. Krebs, "DEOS," Gunter's Space Page, [Online]. Available: https://space.skyrocket.de/doc_sdat/deos.htm. [Accessed 10 January 2023].

[5] H. Benninghoff, F. Rems, E.-A. Risse, P. Irmisch, B. Brunner, M. Stelzer, R. Lampariello, R. Krenn, C. Stangl, R. Faller, O. Peinado and M. Reiner, "RICADOS - Rendezvous, Inspection, Capturing and Detumbling by Orbital Servicing," *7th International Conference on Astrodynamics Tools and Techniques,* 2018.

[6] C. Stangl, B. Lotko, M. Geyer, M. Oswald and A. Braun, "GECCOS – the new Monitoring and Control System at DLR-GSOC for Space Operations, based on SCOS-2000," *SpaceOps 2014 - 13th International Conference on Space Operations,* 2014.

[7] R. Bischoff, J. Kurth, G. Schreiber, R. Köppe, A. Albu-Schäffer, A. Beyer, O. Eiberger, S. Haddadin, A. Stemmer, G. Grunwald and G. Hirzinger, "The KUKA-DLR Lightweight Robot arm – a new reference platform for robotics research and manufacturing," *ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics),* 2010.

[8] Consultative Committee for Space Data Systems (CCSDS), CCSDS 131.0-B-4; TM Synchronization and Channel Coding - Blue Book.

[9] Consultative Committee for Space Data Systems (CCSDS), CCSDS 132.0-B-3; TM Space Data Link Protocol - Blue Book, 2021.

[10] Consultative Committee for Space Data Systems (CCSDS), CCSDS 133.0-B-2; Space Packet Protocol - Blue Book, 2020.

[11] Consultative Committee for Space Data Systems (CCSDS), CCSDS 231.0-B-4; TC Synchronization and Channel Coding - Blue Book, 2021.

[12] Consultative Committee for Space Data Systems (CCSDS), CCSDS 232.0-B-4; TC Space Data Link Protocol - Blue Book, 2021.

[13] D. Weber, C. Garcia Acero, M. Gnat, A. Hauke and F. Huber, "End to end simulation of On-Orbit-Servicing: Implementation of communications," *American Institute of Aeronautics and Astronautics,* 2018.

[14] A. Hauke and E. Barkasz, "Multi-Mission Support with WARP," *SpaceOps 2012 Conference,* 2012.

[15] The Tcpdump Group, "tcpdump," [Online]. Available: https://www.tcpdump.org/. [Accessed 27 January 2023].

[16] wireshark.org, "Wireshark," wireshark.org, [Online]. Available: https://www.wireshark.org/. [Accessed 27 January 2023].

[17] Software in the Public Interest, Inc., "tcpdump - dump traffic on a network," [Online]. Available: https://manpages.debian.org/testing/tcpdump/tcpdump.8.en.html. [Accessed 27 January 2023].

[18] D. Weber, R. Falcone, M. Gnat, A. Hauke and F. Huber, "Technical studies for operations with real-time communications in robotic missions," *American Institute of Aeronautics and Astronautics,* 16 05 2016.

[19] Linux Kernel Organization, Inc., "High resolution timers and dynamic ticks design notes," [Online]. Available: https://www.kernel.org/doc/Documentation/timers/highres.txt. [Accessed 27 January 2023].

[20] The Tcpdump Group, "Miscellaneous Information Manual," 14 July 2020. [Online]. Available: https://www.tcpdump.org/manpages/pcap-tstamp.7.txt. [Accessed 27 January 2023].

[21] VMware, Inc., "Latency Sensitivity," [Online]. Available: https://docs.vmware.com/en/VMware-vSphere/7.0/com.vmware.vsphere.resmgmt.doc/GUID-9F4FD589-A73B-454A-A5A5-FED4C0F918C3.html. [Accessed 27 January 2023].

[22] The Network Time Protocol Project, "HOWTO: Understanding and using the Network Time Protocol," [Online]. Available: http://www.ntp.org/ntpfaq/NTP-s-trbl-spec.htm#AEN5674. [Accessed 27 January 2023].

[23] International Telecommunication Union, "Y.1540 : Internet protocol data communication service - IP packet transfer and availability performance parameters," 09 03 2020. [Online]. Available: https://www.itu.int/rec/T-REC-Y.1540-201912-I/en. [Accessed 26 01 2023].

[24] The Open Group, "The Open Group Base Specifications Issue 7," 2018. [Online]. Available: https://pubs.opengroup.org/onlinepubs/9699919799/basedefs/V1_chap04.html#tag_04_16. [Accessed 27 January 2023].