International Conference on Formal Methods
and Foundations of Artificial Intelligence
Eszterházy Károly Catholic University
Eger, Hungary, June 5–7, 2025

Benchmarking Data Logging Strategies in ROS-Integrated Multi-Sensor Robots Under Network Constraints

Nour Elhouda Ben Saadi[a], **Zoltan Istenes**[b]

[a]Eotvos Lorand Univeristy
nourelhoudabensaadi@gmail.com

[b]Eotvos Lorand Univeristy
istenes@inf.elte.hu

Abstract

Efficient data logging is critical in robotic systems, especially under constrained network conditions where local storage expansion is impractical. This work benchmarks multiple logging strategies—including JSON serialization, JSON Lines (JSONL), WebSocket-based rosbridge, native ROS TCP/UDP transports, and NFS-based recording—on a multi-sensor robot over a 100 Mbps constrained network. We evaluate dropped messages, resource usage, and storage footprint across high-throughput topics such as LiDAR, RGB-D images, IMU, and GNSS data. Results show that TCPROS achieves the highest reliability, while UDPROS, rosbridge, and JSON-based methods face significant losses under saturation. NFS recording offers good performance when network stability is maintained. Our findings reveal key trade-offs between transport guarantees, network limitations, and message size fragmentation. Future work will explore advanced compression and migration to DDS-based ROS2 systems for improved resilience.

1. Introduction

Reliable data logging is essential in robotic systems, particularly for applications involving sensor fusion, mapping, and autonomous navigation. Robots equipped with high-frequency sensors such as LiDARs, RGB-D cameras, and IMUs generate substantial data streams that must be logged efficiently for downstream processing

and diagnostics. While local storage (e.g., onboard SSDs) is often effective, it is not always feasible in field robotics due to size, weight, power, or environmental constraints [6]. In many deployments, data must instead be logged over bandwidth-limited network links.

Existing tools such as `rosbag` [1] and `rosbridge` [2] enable data logging in ROS systems, yet their performance under constrained networks remains underexplored. Prior studies have evaluated ROS communication [4] and transport protocols [5], but few have addressed the challenges of logging high-bandwidth multi-sensor streams over limited Ethernet connections.

High-throughput deployments face several bottlenecks, including CPU load, RAM saturation, network congestion, and transport-specific issues such as UDP fragmentation losses [5] and TCP-induced queuing delays [3]. These factors particularly affect large messages like images and point clouds, where serialization format and transport protocol critically influence reliability.

In this work, we benchmark several data logging strategies—including JSON serialization, JSONL buffering, WebSocket-based `rosbridge`, native TCP/UDP transports, and NFS logging—on a ROS-integrated multi-sensor robot over a 100 Mbps constrained network. Our experiments analyze message integrity, resource usage, and loss patterns across diverse topics. To our knowledge, this is among the first systematic evaluations of data logging trade-offs under real-world network constraints.

2. Experimental Procedure

Each logging strategy was evaluated through a series of three independent 120-second recording sessions to ensure consistency and statistical robustness. For methods that clearly exhibited early limitations—such as unbuffered JSON logging leading to RAM exhaustion and severe message loss—testing was terminated after identifying the critical bottleneck.

Where applicable, parameter tuning was employed to push the performance limits of each method. For JSONL logging, different buffering strategies were tested to optimize stability. For NFS-based recording, both synchronous and asynchronous write modes were evaluated. Compression settings were selectively applied to investigate whether bandwidth optimization could compensate for underlying transport or serialization bottlenecks.

While compression helped reduce the network burden in several cases, it was insufficient to fully mitigate losses for extremely large data types such as LiDAR point clouds. Future work will explore advanced compression schemes, such as Octree-based point cloud compression, to further improve data logging reliability in constrained environments.

Compatibility issues arising from the technologies used—such as base64 encoding overheads in rosbridge or fragmentation vulnerability in UDP—were carefully noted and factored into the comparative evaluation.

3. Results and Discussion

Table 1 summarizes the benchmarking results across the different logging methods. Native TCPROS logging inside a Docker container achieved the highest overall message integrity, with almost no loss except for very high-bandwidth topics such as images and point clouds. Compression strategies helped mitigate occasional losses in these cases.

UDPROS showed good performance overall but suffered random losses on large fragmented messages due to UDP's lack of delivery guarantees. WebSocket-based rosbridge, even with compatibility fixes, exhibited moderate message integrity with higher CPU consumption, primarily due to base64 serialization overheads.

Buffered JSON Lines (JSONL) logging performed more reliably than raw JSON serialization, but remained less efficient than native ROS transports. Raw JSON logging led to rapid RAM saturation, unstable behavior, and significant message truncation, making it unsuitable for extended logging.

NFS-mounted recording experienced significant message loss despite moderate CPU usage, largely due to network delays and NFS protocol overhead, emphasizing the importance of stable high-speed networks for remote disk writing.

Table 1. Summary of Logging Method Performance

Method	**CPU**	**Integrity**	**Disk**	**Notes**
TCPROS (Docker)	Medium	Very High	Normal	Best integrity, minor compression losses
UDPROS (Docker)	Medium	High	Normal	Some losses on large msgs, mitigated by compression
NFS mount	Lower	Poor	Normal	Slower writes, sensitive to network stability
WebSocket (Docker)	Higher	Moderate	Higher	Serialization overhead, more CPU usage
JSONL (16K buffer)	Higher	Moderate	Higher	Safer than JSON, some base64 overhead
JSON raw	Very High (RAM)	Poor	Huge	RAM exhaustion, early logging failure

Overall, the experiments highlight the trade-offs between CPU load, message integrity, and disk usage under constrained network conditions. Native ROS transports (TCPROS and UDPROS) remain the most reliable solutions, with compression offering additional resilience for high-bandwidth sensor data. Future work will focus on advanced compression strategies, such as Octree-based LiDAR data compression, and transition toward DDS-based ROS2 architectures with adaptive QoS to further enhance logging robustness in constrained environments.

References

[1] O. S. R. Foundation: *rosbag - ROS Wiki*, http://wiki.ros.org/rosbag, 2024.

[2] O. S. R. Foundation: $rosbridge_suite - ROSWiki$, http://wiki.ros.org/rosbridge_suite, 2024.

[3] X. Li, B. He, J. Wang: *TCP Delay Analysis in High Bandwidth-Delay Networks for Robotics*, IEEE Access 8 (2020), pp. 174920–174930.

[4] Y. Maruyama, S. Kato, T. Azumi: *Exploring the Performance of ROS Communication*, Proceedings of the 13th IEEE International Conference on Embedded Software and Systems (2016), pp. 1–8.

[5] J. Morales, A. Gil, O. Reinoso: *Evaluation of Communication Protocols for ROS-Based Mobile Robots*, Sensors 19.15 (2019), p. 3361.

[6] S. Yaghoubi, G. Dissanayake: *Field Robotics: From Systems to Deployment*, Annual Review of Control, Robotics, and Autonomous Systems 5 (2022), pp. 165–189.